

# Web İçerik Geliştirme Platformu

# VFABRİKA PROGRAMI KULLANIM KILAVUZU

Bu doküman VFabrika programının nasıl kullanılacağını anlatır. VFabrika, Sebit Eğitim ve Bilgi Teknolojileri A.Ş. bünyesinde üretilen bir Windows uygulamasıdır.



Sebit Eğitim ve Bilgi Teknolojileri A.Ş. 2020. Her Hakkı Saklıdır.

# İçindekiler

Giriş5
Açılış Ekranı6
Yeni Proje Oluşturma 6
VFabrika Sürüm Bilgisi ve Güncellemeler8
VFabrika Geliştirme Ortamı
Dizayn ve Bloklar10
Timeline (Zaman Çizelgesi)11
Çözüm Yöneticisi ve Hazır Materyaller14
Sahne Nesneleri
Özellikler ve Hizalama15
Araç Çubuğu15
Interaction (Etkileşim)17
BÖLÜM 1- VFABRİKA İLE KODLAMAYA GİRİŞ 18
Değişkenler18
Variable (Değişken) nedir?18
Veri Tipleri
Değişken Kapsama Alanı19
Get – Set Erişimcileri
Operatörler
Operatör nedir?
Aritmetik Operatörler
Mantıksal Operatörler 21
Karşılaştırma Operatörleri
Karar Kontrol Yapıları
If Then (Eğer ise)
If Then Else (Eğer ise Değilse)
If Then Else Return (Eğer ise Değilse Döndür) 27
Switch Case (Değişken Durum)27
Döngüler (Loops)
Foreach Döngüsü (Her Eleman İçin Döngüsü) 30



	While Döngüsü (Sürekli Yap Döngüsü)	. 31
	Do While Döngüsü (Yap ve Doğruysa Sürekli Yap Döngüsü)	. 31
	Break (Durdur)	. 32
	Continue (Devam Et)	. 32
Li	steler	. 35
	Create List (Liste Oluşturma)	. 35
	Create From Comma Text (Virgülle Ayrılmış Metinden Liste Oluştur)	. 36
	Add Item (Öge Ekleme)	. 36
	Add Item Into (Sıraya Öge Ekle)	. 36
	Remove Item (Öge Çıkarma)	. 37
	Remove Item At (Pozisyondaki Ögeyi Sil)	. 37
	Length (Liste Uzunluğu)	. 37
	Contains (Mevcut)	. 37
	Get Item At (Poziyondaki Ögeyi Oku)	. 38
	Index Of (Pozisyon Bul)	. 38
	Set Item At (Pozisyondaki Ögeyi Değiştir)	. 38
	Shuffle List (Listeyi Karıştır)	. 38
	Sort (Sırala)	. 38
Ρ	rosedürler	. 40
	Prosedür Nedir?	. 40
	Procedure (Prosedür Oluşturma)	. 40
	Call (Çağırma)	. 40
	Return Result (Sonuç Çağırma)	. 41
	Call for Result (Sonuç Döndüren Prosedür Çağırma)	. 41
S	tring (Metin)	. 43
	Compare Text (Karşılaştır)	. 43
	Contains (İçeriyor mu)	. 43
	Is Empty (Boş mu)	. 44
	Join (Ekle)	. 44
	Index of Text (Metnin Konumunu Bul)	. 44
	Length (Uzunluk)	. 44



Replace All (Hepsini Değiştir)	
Segment (Böl)	
Starts At (Başlangıç Pozisyonunu Bul)	
Trim (Boşlukları Temizle)	
To Lowercase (Küçük Harfe Çevir)	
To Uppercase (Büyük Harfe Çevir)	
To Text (Metne Çevir)	
Concat (Birleştir)	
Split (Ayır)	
BÖLÜM 2 - FORM NESNELERİ	
Button (Buton)	
Image Button (Resim Butonu)	
Radio Button (Seçenek Butonu)	50
CheckBox (Kontrol Butonu)	50
Dropdown List (Açılan Kutu)	
Textbox (Metin Kutusu)	
Slider (Sürgü)	
BÖLÜM 3 – MEDYA NESNELERİ	
Image (Resim)	
Sound (Ses)	55
Video	55
BÖLÜM 4 – TASARIM NESNELERİ	57
Canvas (Resim Kâğıdı)	57
Content (İçerik)	61
Table (Tablo)	62
Highlight (Vurgu)	64
Primitive Objects (Temel Nesneler)	64
Smart Objects (Akıllı Nesneler)	65
Parallax Scroll (Paralaks Kaydırma)	65
Polygon (Çokgen)	
BÖLÜM 5 – ARAÇLAR	68

Sebit Eğitim ve Bilgi Teknolojileri A.Ş. 2020. Her Hakkı Saklıdır.



Delay (Erteleyici)
Timer (Zamanlayıcı)
Trigger (Tetikleyici)
BÖLÜM 6 – SEBİT FRAMEWORK
Sebit Framework
Feedbacks (Bildirimler)71
Questions (Sorular)72
Skip Intro (Tanıtımı Geç)
Animation Popup (Animasyon Penceresi)74
Hypothesis (Hipotezler)
BÖLÜM 7 – HAZIR KÜTÜPHANE FONKSİYONLARI
Math (Matematik)77
Math Advanced (İleri Matematik)
Trigonometry (Trigonometri)
Json Data Set
BÖLÜM 8 – DİĞER ARAÇLAR
General
Animation
BÖLÜM 9 – ŞABLONLAR
Kare Bulmaca
Boşlukları Doldurma (Açılır Kutular)91
Boşlukları Doldurma (Metin Kutular)93
Dönen Kartlar Projesi
Kelimeyi Tahmin Et Projesi
Boşluklarla Eşleştirme Projesi
Görseller Üzerinde Eşleştirme Projesi
Çoktan Seçmeli Projesi 101
Cümle Sıralama Projesi 101
Dönen Tekerlek Projesi 102



# Giriş

VFabrika, öğretmenlerin farklı becerilere sahip öğrenci gruplarıyla birlikte, kolaylıkla özgün etkileşimli içerik üretmelerini sağlayan "HTML5 İçerik Geliştirme Platformu"dur.

Öğretmenlerin ve öğrenci gruplarının alıştırma, deney, canlandırma, eğitsel oyun gibi öğrenme materyallerini kod yazmadan, herhangi bir yardım almadan, nitelikli bir şekilde üretmeleri için tasarlanmıştır.

VFabrika'yla öğretmenlerin liderliğindeki öğrenciler, üretilecek e-içerik için planladıkları çalışmaları, kendi yetenek ve ilgileri doğrultusunda tüm süreçlere katkıda bulunarak iş birliği içerisinde tamamlarlar.

VFabrika, üretim sürecinde öğretmenlere ve öğrencilere sistematik kurgu, tasarım, planlama, algoritmik düşünme ve iş birliği içinde çalışma becerileri kazandırır.

# Açılış Ekranı

VFabrika'yı başlattığımızda karşımıza "Geliştirme Ortamı" gelir. Geliştirme Ortamında sol taraftaki menüde "Yeni Proje Oluştur", "Proje Aç" ve onların altında da "Önceki Ögeler" kısmı yer alır. Önceki ögeler kısmında, daha önce çalıştığımız projeler listelenir. Bu listeden bir projeye tıklayarak kaldığımız yerden devam edebiliriz.



Şekil 1: Açılış Ekranı

# Yeni Proje Oluşturma

Yeni bir proje oluşturmak için, Açılış Ekranında "Yeni Proje..." linkine tıklayarak veya Dosya > Yeni Proje adımını takip etmemiz yeterli olur. Bunlara ek olarak, "CTRL + N" kısayolu ile de yeni bir proje oluşturabiliriz.



#### Sayfa | 7

Yeni Proje			x
Yeni Proje Oluştur Bu ekran ile yeni bir proje oluşturabilirsiniz.			
Web Sablonlar Diğer		Web Projesi On tanımlı boyutlara sahip web projesi.	
Proje adı Projem			
Konum C:\Users\sebit\Desktop		qq	
Çözüm Adı	Projem		
Önceki		Önceki Sonraki İptal	

Şekil 2: Yeni Proje Penceresi

Açılan "Yeni Proje" penceresinde Boş bir genel proje seçebileceğimiz şablon projelerden birini de seçebiliriz. Şablon projelerde VFabrika ekibi tarafından oluşturulmuş hazır ürünleri, verilen seçenekleri belirleyerek istediğimiz şekilde oluşturabiliriz.

Proje şeklimizi seçtikten sonra aşağıda yer alan projenin adını, konumunu ve çözüm adını belirleyerek devam edebiliriz. Bir sonraki ekranımızda projemizin boyutlarını gösteren pencere gelir. Standart boyutlar 762 x 463'tür. Bu boyutları dilediğiniz gibi değiştirebilirsiniz.

Yeni Proje	x	3
Yeni Proje Oluştur         Bu ekran ile yeni bir proje oluşturabilirsiniz.		
Width 762 + px (Maximum 4000 px) Height 463 + px (Maximum 4000 px)		
Önceki Oluştur İptal		

Şekil 3: Yeni Proje Penceresi

# VFabrika Sürüm Bilgisi ve Güncellemeler

Kullandığımız VFabrika sürüm bilgisini Yardım > Hakkında yolunu izleyerek karşımıza çıkan pencereden öğrenebiliriz. Bu pencerede VFabrika sürümünü, Donanım Anahtarını ve Lisans bilgilerini öğrenebiliriz. VFabrika'nın sürüm güncellemesini kontrol etmek için de Yardım > Güncellemeleri Kontrol Et yolunu kullanabiliriz.

Hakkında	x
<b>V</b> Fabrika	
Versiyon 7.0.6 Beta 8 Her hakkı saklıdır. 2020 Sebit	
Web	
Lisans Eklentiler Test Kapat	

Şekil 4:Hakkında Penceresi



# VFabrika Geliştirme Ortamı

VFabrika'da projelerimizi geliştirme ortamında oluştururuz. VFabrika geliştirme ortamı çok kullanışlı ve anlaşılabilir şekilde tasarlanmıştır. Temel olarak 6 bölümden oluşmaktadır. Bunlar; Dizayn ve Bloklar, Zaman Çizelgesi, Çözüm Yöneticisi ve Hazır Materyaller, Sahne Nesneleri, Özellikler ve Hizalama, Araç Çubuğu. Bu bölümler sayesinde VFabrika'yı daha verimli kullanabilirsiniz.



Şekil 5: Geliştirme Ortamı

Varsayılan Geliştirme Ortamı: **A**. Dizayn ve Bloklar **B**. Timeline (Zaman Çizelgesi) **C**. Çözüm Yöneticisi ve Hazır Materyaller **D**. Sahne Nesneleri **E**. Özellikler ve Hizalama **F**. Araç Çubuğu

#### Dizayn ve Bloklar

Dizayn çalışma alanı, sahne nesnelerini projeye yerleştirebilmemizi ve görsel ayarlama yapmamızı sağlar. Dizayn araç çubuğundaki nesneleri sürükle – bırak yöntemiyle sahneye ekleyebiliriz. Sahne sınırlarının dışında olan görseller projeyi çalıştırdığımızda ekranda görünmez.

Blok çalışma alanında projemizi kodlayabilir, sahneye eklediğimiz nesnelerin özelliklerini bloklar yardımıyla değiştirebilir ve çeşitli algoritmalar oluşturabiliriz.



# Timeline (Zaman Çizelgesi)

VFabrika'da Timeline (Zaman çizelgesi); frame (kare) ve layer'lardan (katman) oluşan ve bunları kontrol eden bir yapıdır. Animasyonları kareler aracılığı ile zaman çizelgesinde oluşturabiliriz. VFabrika'da süre uzunlukları karelere bölünür ve kare sayısına göre animasyonların ne kadar hızlı veya yavaş olacağını ayarlanabilir. Projede eğer blok kullanılmışsa kullanılan katmanın zaman çizelgesine "b" işareti gelir. Bu işaret, o katmanda blokların da olduğunu gösterir.



Şekil 6: Zaman Çizelgesi

Layer (Katman)

	۲	
🔣 Katman 3 🛛 🖌	•	+
🔤 Katman 2	+	•
🔟 Katman 1	+	•
📃 🗑		▼
Şekil 7: Katman		

Katman, projedeki nesneleri gruplandırmamızı sağlayan ve animasyonları daha kolay getiren bir zaman çizelgesi bileşenidir. Katmanlarda sıralama önemlidir. En üstte yer alan katman sahnenin en önünde bulunurken en alttaki katman da sahnenin en gerisinde bulunur.

Katman panelinden kolaylıkla yeni katman ekleyebilir, var olan katmanı silebilir, katmanların yerini değiştirebiliriz. Ayrıca katmanları kilitleyebilir veya sahnedeki görünürlüğünü kapatabiliriz.

Bir katmanı kilitlediğimizde, katmandaki nesneleri de kilitlemiş oluruz. Böylece o katmandaki nesneleri sahnedeyken seçemeyiz veya yerini değiştiremeyiz. Örneğin, projemizde bir arka plan olsun ve onu da "arka plan" katmanına koymuş olalım. Arka planı ekledikten ve ayarladıktan sonra çalışmamız bitince onu kilitlemek bizim için faydalı olabilir. Böylece yerini değiştirmek, silmek gibi yanlışlıkla yapılabilecek hatalardan da kaçınmış oluruz.

Kilit düğmesinin yanında bulunan göze tıklayarak seçtiğimiz bir katmanı gizleyebiliriz. Gizlediğimiz katmanda bulunan nesneler sahnede görünmez.



#### Frame (Kare)

VFabrika'da süre uzunlukları kare olarak oluşturulur. Varsayılan olarak 1 saniyedeki kare sayısı 24'tür. Ancak bunu değiştirmek mümkündür. Bunun için, araç çubuğundaki Proje > Özellikler yoluyla karşımıza çıkan pencereden yararlanabiliriz. Animasyonları da bu karelerle oluşturabiliriz. Kare eklemek için F5 kısa yolunu kullanabiliriz. Sahnedeki tüm nesneler eklediğimiz karede de aynı şekilde yer alır.

#### Keyframe (Anahtar Kare)

Anahtar kare, sahnede veya zaman çizelgesinde bir değişikliğin meydana geleceği karelerdir. Örneğin, yolda giden bir araba animasyonu yapmak istediğimizde, arabanın başlangıç noktası ilk kare olurken arabanın sahnede en son bulunacağı 20. kareye bir anahtar kare ekleriz ve sahnemizdeki araba nesnesini de en son duracağı yere taşırız. Ardından, 1 ile 20. kare arasında herhangi bir yerde sağ tıklayarak "Hareket Oluştur" dediğimizde arabanın belirtilen doğrultuda hareket ettiğini göreceğiz. Anahtar kare eklemek için F6 kısayolunu kullanabiliriz.

#### Blank Keyframe (Boş Anahtar Kare)

Boş anahtar kare, eklediğimiz karede sahneyi boş bırakmak için kullanılır. Anahtar kare eklediğimizde, sahnedeki tüm nesneler aynı şekilde eklendiği kareye taşınır fakat boş anahtar karede sahne hiçbir nesne yer almaz. Boş anahtar kare eklemek için F7 kısayolunu kullanabiliriz.

Timeline Player Control	Timeline Player Control (Zaman Çizelgesi Kontrolü) bloğu, zaman çizelgesini başlatma veya durdurma komutları vermemizi sağlar.
Timeline Go To And Timeline Player Control Go To	Timeline Player Control Go To (Zaman Çizelgesi Kontrol ve Kare Değişimi) bloğu, zaman çizelgesinde belirtilen kareye gidilmesini ve o karedeyken oynatma veya durdurma komutlarını vermemizi sağlar.

#### Timeline (Zaman Çizelgesi) Blokları



Set Fps to Timeline Set Fps	Timeline Set Fps (Saniyedeki Kare Sayısını Değiştir) bloğu, zaman çizelgesinde saniyedeki kare sayısını değiştirmemizi sağlar.
Timeline reset	Timeline Reset (Zaman Çizelgesini Sıfırla) bloğu, zaman çizelgesini sıfırlamamızı sağlar.
Timeline Event Initialize Timeline Event	Timeline Event (Zaman Çizelgesi Olayı) bloğu, zaman çizelgesi başladığında (initialize), kare değiştiğinde (frame change) veya durum değiştiğinde (state change) neler yapılacağını belirlememizi sağlar.
Get Timeline Frame Index Get Frame Index	Get Frame Index (Kare Indeksini Al) bloğu, zaman çizelgesinde hangi karede yer aldığımızı öğrenmemizi sağlar.
Get Timeline State	Get Timeline State (Zaman Çizelgesi Durumunu Al) bloğu, zaman çizelgesi oynuyorsa True (Doğru) oynamıyorsa False (Yanlış) değerini döndürür.



Get Layer Visibility	Get Layer Visibility () bloğu, ismini yazdığımız katman eğer görünür ise True (Doğru) görünür değilse False (Yanlış) değerini döndürür.
Set Layer Visibility	Set Layer Visibility () bloğu ismini yazdığımız katmanın görünürlüğünü açmamızı ya da kapamamızı sağlar. True (Doğru) değeri ile açar, False (Yanlış) değeri ile kapatabiliriz.



#### Çözüm Yöneticisi ve Hazır Materyaller

Çözüm yöneticisi paneli aracılığıyla projenin dosya sistemine ulaşabiliriz. Projemizde yer alan tüm klasörler burada sıralanır. Varsayılan olarak datasets, images, scripts, sounds, styles, videos klasörleri yer alır. Çözüm yöneticisinde aynı zamanda birden çok proje dosyası ile çalışabiliriz. Bunun için en üstte yer alan "Çözüm" başlığına sağ tıklayarak Ekle > Yeni Proje ya da Ekle > Varolan Proje yollarını izleyerek ekleyebiliriz.

Hazır Materyaller bölümünden de proje için kullanabileceğimiz resimlere ulaşabiliriz. Bunun için, arama kısmındaki metin kutusuna anahtar kelimeleri yazdıktan sonra çıkan sonuçlardan dilediğimizi sürükle bırak yöntemiyle sahneye ekleyebiliriz.

Şekil 8: Çözüm Yöneticisi ve Hazır Materyaller



### Sahne Nesneleri



Sahnede kullanılan nesneleri hiyerarşik yapısına göre bu panelde bulabiliriz. Katman sırası değiştiğinde bu panelde de katman sırası değişmektedir. Kullanılan tüm nesnelere ulaşabilir ve hiyerarşik sırasını bu panelle değiştirebiliriz. Sahnede bir nesne seçildiğinde, eğer nesne görünür durumdaysa ekranda seçili hâle gelir ve nesnenin içinde bulunduğu Layer ve Keyframe otomatik olarak seçilir.

### Özellikler ve Hizalama

Ozellikler 🗖 म 🗙				
Ball gu button1				
Davranış	^			
Döndürme	0			
El İşareti Göster	False			
Değer	ОК			
Genel	^			
Opaklik	100			
Özellikler	(Collection)			
Sinif Adi				
Aktif	True			
▶ Konum	295; 326			
Ad	button1			
Boyut	100; 25			
Biçimler	(Collection)			
Etiket				
Type Name	inputButtonDesignO			
Görünür	True			
Özellikler Hizalama				

Şekil 10: Özellikler ve Hizalama

VFabrika'da her nesnenin kendine ait birtakım özellikleri vardır. Bu özelliklere ulaşmak için, sahne panelinden veya sahnedeyken nesneye tıklamamız yeterlidir. Böylece o nesneye ait özellikleri bu panelde bulabilir ve değiştirebiliriz.

Genel özellikler başlığı her nesnede bulunan; opaklık, konum, ad, boyut, görünür, aktif gibi özellikleri içerir.

**NOT:** Biçimler kısmıyla, seçili nesne için CSS özellikleri ekleyebiliriz.

Hizalama panelinde hizala, dağıt, boyut eşitle gibi özellikler bulunur. Bu özelliklerle seçili nesnenin hizalama işlemlerini yapabiliriz. Ayrıca, sahneye göre hizala seçeneği ile otomatik olarak hizalanmasını da sağlayabiliriz.

# Araç Çubuğu

Dosya Düzenle Gorünüm ZarmanÇizelgesi Proje Yayınla Eğitim Sebit Araçlar

Şekil 11:Araç Çubuğu

VFabrika programında kullanılan temel araçları barındıran paneldir. Panelde şu araçlar bulunur: Yeni Proje, Proje Aç, Projeyi Kaydet, Projeyi Farklı Kaydet, Geri Al, İleri Al, Tercihler, Kopyala,



Web Tarayıcı Varsaylan 💌 Çalışma Düzeni •

Kes, Yapıştır, Proje Ayarları, Yayınla, Yayınla ve Çalıştır, Yayın Ayarları, Zaman Çizelgesini Oynat, Hakkında.

Tarayıcı kısmıyla projeyi çalıştırdığımız zaman hangi tarayıcının kullanılacağını seçebiliriz. Varsayılan olarak seçiliyse bu durumda bilgisayarın varsayılan tarayıcısı kullanılır.

En sağda yer alan Çalışma Alanı bölümünde programın arayüzünü düzenleyebilir, düzenlediğimiz arayüzü kaydedebilir ve daha önce kaydettiğimiz arayüzleri yükleyebiliriz. Çalışma Alanı menüsünden "default" seçeneğine tıkladığımızda, program arayüzünü varsayılan hâline geri döndürür.



# Interaction (Etkileşim)

VFabrika'da oluşturduğumuz içerikler yalnızca tasarım ve sunumdan meydana geliyorsa bu projeler statik (durağan) olarak nitelendirilir. Statik projelerde kullanıcı projeyi kullanırken yalnızca izleyicidir. Herhangi bir şekilde projeye veya içeriğine müdahale edemez. Ancak, etkileşim nesnelerini kullanarak oluşturacağımız projeyi dinamik hâle getirebiliriz. Böylelikle kullanıcı ile etkileşimli bir proje oluşturulabilir.

Enable User Interaction Enable User Interaction	Enable User Interaction (Kullanıcı Etkileşimini Etkinleştir) bloğu, kullanıcı etkileşiminin aktif hale getirilmesini sağlar.
Disable User Interaction Disable User Interaction	Disable User Interaction (Kullanıcı Etkileşimini Devre Dışı Bırak) bloğu, kullanıcı etkileşimi aktif ise devre dışı bırakarak etkileşimi kapatır.
Do Con Input Key Event	On Input Key Event (Tuş Olayı) bloğu, kullanıcı klavyeden bir tuşa bastığında neler yapılacağını belirlememizi sağlar.

**NOT:** Tuş olayında Key Up, Key Down, Key Press olmak üzere üç seçenek vardır. Key Up, tuşa basma işlemi sonlandığında; Key Down, tuş basılı iken ve Key Press ise tuşa basıldığında çalışır.

Open Html Link New Tab Mttp:// Open Html Link	Open Html Link (Html Link Aç) bloğu, girilen web sayfası linkinin açılmasını sağlar.
Character from Char Code Character from Char Code	Character from Char Code (Char Code'dan Karakter Getir) bloğuna karakter kodunu eklediğimizde, eklenilen karakter kodunun aktif olması sağlanır.
Stage Mouse Event Mouse Click On Stage Mouse Event	On Stage Mouse Event (Sahnede Fare Olayı) bloğu, fareye tıklandığında neler yapılacağını belirlememizi sağlar.
Is Key Pressed KeyCode Is Key Pressed	ls Key Pressed (Tuşa Basıldı mı) bloğu, eklenen karakter kodunun basılıp basılmadığını kontrol eder.

Sebit Eğitim ve Bilgi Teknolojileri A.Ş. 2020. Her Hakkı Saklıdır.



# BÖLÜM 1- VFABRİKA İLE KODLAMAYA GİRİŞ Değişkenler

### Variable (Değişken) nedir?

Verilerin, bilgisayarın hafızasında tutulduğu alana verilen isimdir. Değişkenleri bir kutu gibi düşünebiliriz. İçine bir değer ekleyebilir, değeri değiştirebilir veya kaldırabiliriz. Değişkenler için bilgisayarın geçici hafızasında (RAM) bir yer ayrılır. Değişkenlere atadığımız isimler sayesinde onları hafızadan çağırabilir ve üzerinde işlemler yapabiliriz.

VFabrika'da bir değişken tanımlayalım;





Bilgisayar hafızasında *degiskenim* adlı bir kutu oluşturduk ve içine de 5 değerini koymuş olduk. Değişkenin değerine ulaşmak için ona atadığımız isim kullanılır.

#### Veri Tipleri

VFabrika'da temel olarak üç çeşit veri tipi vardır. Bunlar; sayı, metin ve boolean.

otomatik olarak tam sayıya dönüşür.



Sayı veri tipinde, tam sayı ve ondalık sayı değerleri girilebilir.

Şekil 13



String (Metin) veri tipinde metin, cümle, harf değerleri girilebilir.



Şekil 15

Boolean ikili veri tipidir ve yalnızca iki değer bulunur. Bunlar doğru ve

Tam sayı değerine ondalık bir değer giremeyiz. Ondalık girdiğimiz değer,

Sebit Eğitim ve Bilgi Teknolojileri A.Ş. 2020. Her Hakkı Saklıdır.

yanlıştır.



Bazı programlama dillerinde değişken tanımlamadan önce veri tipini belirtmek zorunludur. VFabrika'da değişken tanımlarken veri tipini belirtmek gerekmemekte, program bunu otomatik olarak algılayabilmektedir.

#### Değişken Kapsama Alanı

Kapsama alanı, tanımlanan değişkenin programın hangi bölümünde çalışabileceğini belirler. VFabrika iki farklı kapsama alanına sahiptir. Bunlar Yerel (Local) ve Genel (Global) kapsama alanlarıdır.

Local (Yerel) Kapsama Alanı: Yerel tanımlanmış değişkenler yalnız kullanıldığı blok veya fonksiyon içinde tanınabilir ve kullanılabilirler. Programın tamamında kullanılamazlar. Global (Genel) Kapsama Alanı: Genel tanımlanan değişkenler programın her yerinde tanınabilir ve kullanılabilirler. Sekil 10 Initialize global (MyVar) to Initialize Global

#### Get – Set Erişimcileri

Get (Değişken Oku) ve Set (Değişken Ayarla) erişimcileri, değişken tanımladıktan sonra değişkene ulaşmamızı ve üzerinde değişiklik yapmamız sağlar. Dolayısıyla, değişken ile ilgili tüm işlemleri bu erişimciler aracılığıyla yapabilmekteyiz.

**Get** erişimcisi ile tanımlanmış olan değişkeni çağırabiliriz. Böylelikle değişkene atanmış değere ulaşabiliriz.

Set erişimcisi ile tanımlanmış olan değişkenin değerini ve veri tipini değiştirebiliriz. Örneğin; başlangıçta 5 olarak tanımlanmış bir değişkenin değerini 10 yapabiliriz.





Şekil 19



**Uygulama 1:** Tam sayı değerine sahip genel bir değişken tanımlayıp butona bastığımızda değerini değiştiren bir uygulama yapalım.

\*Buton ve metin kutusunun özellikleri ilerleyen bölümlerde daha detaylı anlatılacaktır.

initialize global degiskenim to 5	
When button1 . Mouse Click set degiskenim to Vitamin " Do Set text1 . Text to get degiskenim	

Şekil 20

- 1- *degiskenim* isimli bir değişken tanımlayarak değerini 5 olarak belirledik.
- 2- Butona tıklandığında, değişkenimizin değerini set erişimcisini kullanarak *Vitamin* olarak değiştirdik.
- 3- Get erişimcisi ile değişkenimizi çağırdık ve metin kutusuna değişkenimizin yeni değerini yazdırdık.

	Buton	
Vita	amin	

Şekil 21

Şekil 10'da görüldüğü üzere, butona tıkladığımızda ekrana 5 değerini değil, *Vitamin* değerini yazdırdık.

Sebit Eğitim ve Bilgi Teknolojileri A.Ş. 2020. Her Hakkı Saklıdır.



# Operatörler

# Operatör nedir?

Operatörler, programlama dillerinde matematiksel, mantıksal ve ilişkisel işlemleri yapmamızı sağlayan, aynı zamanda tek başına kullanılamayan sembol ya da karakterlere denir. Operatörlerin işlem yapabilmesi için gerekli olan değişken veya sabit değerlere ise **operand** denir. Örneğin, 4 + 5 işlemini ele alırsak, burada rakamlara "operand" toplama işaretine ise "operatör" denir. VFabrika'da yer alan operatör tipleri şunlardır:

- Aritmetik Operatörler
- Mantıksal Operatörler
- Karşılaştırma Operatörleri

### Aritmetik Operatörler

Matematiksel işlemler aritmetik operatörler vasıtası ile yapılır. VFabrika'da dört temel aritmetik operatör bulunmaktadır.



Şekil 22: Toplama

Şekil 23: Çıkarma

Şekil 25: Çarpma

Şekil 24: Bölme

Bunlara ek olarak, diğer programlama dillerinde karakter olarak bulunan Mod alma (%) VFabrika'da karakter olarak değil yazı olarak bulunmaktadır.



Şekil 26: Mod Alma

### Mantıksal Operatörler

Belirlenen şartları kontrol eden ve true (doğru) ya da false (yanlış) sonuçları çıkaran operatörlere denir. Mantıksal operatörlerde ve (&&), veya (||), değil (!) bulunur. VFabrika'da bu operatörler sembol olarak değil, yazı olarak yer alır.





And (Ve) operatöründe iki şartın da doğru olup olmadığı kontrol edilir. İki operand da doğru ise sonuç doğru olarak çıkar.

Şekil 27: Ve Operatörü

1. Operand	2. Operand	Sonuç	
Doğru	Doğru	Doğru	
Doğru	Yanlış	Yanlış	
Yanlış	Doğru	Yanlış	
Yanlış	Yanlış	Yanlış	



Or (Veya) operatöründe iki şarttan birinin doğru olup olmadığı kontrol edilir. İki operand'dan en az biri doğru ise sonuç doğru olarak çıkar.

Şekil 28: Veya Operatörü

1. Operand	2. Operand	Sonuç	
Doğru	Doğru	Doğru	
Doğru	Yanlış	Doğru	
Yanlış	Doğru	Doğru	
Yanlış	Yanlış	Yanlış	



Şekil 29: Değil Operatörü Not (Değil) operatöründe ise bir tane operand alınır ve o operandın tersi sonuç olarak çıkar. Alınan operand doğru ise yanlış, yanlış ise doğru sonucu çıkar.



# Karşılaştırma Operatörleri

İsminden de anlaşılacağı üzere, karşılaştırma operatörleri iki operandı karşılaştırmak için kullanılır. Karşılaştırma sonucu da doğru veya yanlış olarak çıkar. Karşılaştırma operatörleri şunlardır;



**Uygulama 1:** İki farklı değişken tanımlayıp ardından operatörlerle işlemler yapalım. Butona basıldığında da metin kutusuna sonucu yazdıralım.



Sayfa **| 24** 

initialize global (Ik_degisken) to 5	
initialize global fikinci_degisken to	
When button1     . Mouse Click       Do     Set text1     . Text       Do     . Text     to       get ilk_degisken     +	]

Şekil 36

- 1- İki farklı değişken tanımladık.
- 2- Aritmetik operatörlerde gördüğümüz toplama operatörünü bloğumuza ekledik.
- 3- Get erişimcisi ile değişkenlerimizin değerlerini aldık ve toplama işlemine yerleştirdik.
- 4- Son olarak da ekrandaki text'imize sonucu yazdırdık.



Şimdi de uygulamada tanımladığımız değişkenlerin eşit olup olmadığına bakalım.



Sayfa **| 25** 

initialize global (Ik_degisken) to (5)		
initialize global (kinci_degisken) to		
When button1 . Mouse Click		
Do	To Text	== V get (kinci_degisken V

Şekil 38

- 1- İki farklı değişken tanımladık.
- 2- Karşılaştırma operatörlerinde gördüğümüz eşitse operatörünü bloğumuza ekledik.
- 3- Get erişimcisi ile değişkenlerimizin değerlerini aldık ve eşitse bloğuna yerleştirdik.
- 4- To Text (Metne Çevir) bloğunu da sonucu metin olarak görmek istediğimiz için bloğumuza ekledik.
- 5- Son olarak da ekrana sonucu yazdırdık.

**NOT:** Metne Çevir bloğunu eklemezsek VFabrika yine değişkenlerin birbirine eşit olup olmadığını kontrol eder fakat bunu text1 kutusuna yazamaz. Programlama dillerinde doğru 1 yanlış da 0 ile temsil edilir. Böyle bir karşılaştırma durumunda da sonuç ya 1 ya 0 olarak çıkar fakat 1 ve 0'ı ekrana yazdıramayız. Bunun yerine To Text (Metne Çevir) bloğunu kullanırız. Böylelikle program gelen sonucun ne olduğuna bakar ve onu metne çevirir. Bizim uygulamamızda da eşitse operatörünün sonucu 0 olarak çıkar ve To Text bloğu bunu false (yanlış) olarak metne çevirir.



E	Buton	
	false	

Çıktıda da görüldüğü gibi, butona bastığımızda metin kutusunda false (yanlış) değerini yazdırdı.

Şekil 39

# Karar Kontrol Yapıları

Programlamada, tıpkı gerçek hayatta olduğu gibi, karşımıza koşul ve ihtimaller çıkabilir. Bu tür durumlarda ne yapılacağına önceden karar vermek gerekir. Karar kontrol yapıları, program içinde şartlara bağlı olarak hangi kod bloklarının çalıştırılıp çalıştırılmayacağını uygulayan yapılardır. VFabrika'da bulunan karar kontrol yapıları şunlardır: If Then (Eğer ise) If Then Else (Eğer ise Değilse), If Then Else Return (Eğer ise Değilse Döndür) ve Switch Case (Değişken Durum).

# If Then (Eğer ise)

If Then en temel karar kontrol yapısıdır. Program içinde bir koşul komutu oluşturmamızı ve o şartın sağlanıp sağlanmadığını kontrol etmemizi sağlar. If kısmında şart, then kısmında da şart sağlandığında ne yapılacağı belirtilir.



Blok üzerindeki çarka tıkladığımızda istenildiği kadar else if (değilse eğer) veya else (değilse) eklenebilir.

#### Şekil 40: If Then

#### If Then Else (Eğer ise Değilse)



Şekil 41: If Then Else

If Then Else oluşturulan şartın sağlanıp sağlanmadığını kontrol eder. If Then bloğundan farklı olarak koşul sağlanmadığında da bir komut oluşturabiliriz. Koşul sağlandığında then, sağlanmadığında else kısmında belirtilen bloklar çalışır.



## If Then Else Return (Eğer ise Değilse Döndür)

If Then Else Return, belirlenen şartın sağlanıp sağlanmadığını kontrol eder ve eğer şart sağlanmıyorsa belirtilen bloğu döndürür.



Şekil 43: If Then Else Return

#### Switch Case (Değişken Durum)



Şekil 44: Switch

Switch Case de aynı If Then gibi karar kontrol yapısıdır fakat farklılıkları vardır. Switch Case'de koşul tanımlamak yerine bir değişkenin durumuna bakarak karar verilir. Örneğin, 1 ile 7 arasında bir sayı değerine sahip değişken tanımlayıp bunun hangi güne karşılık geldiğini gösteren bir program yaptığımızda, her seferinde şartımızı ayrı ayrı yazmak yerine Switch Case'de switch kısmına 1 ile 7 arasında değere sahip olan değişkenimizi koyarız, case (durum) kısmında 1, 2, 3 vs. şeklinde değerlerimizi koyup do kısmında da günleri belirtebiliriz. Böylelikle değişkenimiz 3 ise

Çarşamba, 5 ise program bize Cuma'yı gösterir.

Bloğu eklediğimizde, varsayılan olarak bir tane case gelir fakat bloğun üzerinde bulunan çarka tıklayarak istediğimiz kadar case ekleyebiliriz.

**Default:** Switch Case'de değişkenimiz durumların hiçbiri ile eşleşmediğinde ne yapılmasına gerektiğine default ile karar veririz. Switch Case bloğumuzda varsayılan olarak default gelmemektedir. Ancak çarka tıkladığımızda kolaylıkla ekleyebiliriz.



Şekil 45: Default



Şekil 46: Break

**Break (Durdur):** Değişkenimiz belirtilen durumlardan biriyle eşleştiğinde diğer durumlara bakılmasını istemiyorsak durdur bloğunu kullanırız. Böylelikle değişken Switch Case'den çıkar. Aksi durumda değişken şartın sağlandığı her bloğu kontrol eder.





**Uygulama:** Bir değişken tanımlayalım ve bu değişkenin 15'ten büyük mü küçük mü yoksa 15'e eşit mi olduğunu kontrol edelim.

Şekil 47

- 1- Değeri 12 olan *degiskenim* adlı bir değişken tanımladık.
- 2- Butona tıkladığımız zaman ekrana yazmasını istediğimiz için Mouse Click bloğunu ekledik.
- 3- Kıyaslamayı yapabilmek için If Then kontrol bloğunu ekledik.
- 4- Şartlarımızı oluşturduk, eğer büyük veya küçük değil ise eşit olacağından ötürü son olasılığımızı da yerleştirdik.
- 5- Ekrandaki Text'e yazdırmak istediğimiz için Set Text bloğunu ekledik ve ekrana yazdırdık.



Buton	Çıktıda da görüldüğü gibi, değişkenimize 12 değerini verdik ve if then karar kontrol
Değişkenin değeri 15'ten küçüktür.	yapısıyla kontrol ettirdikten sonra çıkan sonucu ekrana yazdırdık.

**Uygulama:** Bu uygulamamızda da switch case'i kullanarak 1-3 arası bir değişken tanımlayarak onu yazı ile ekrana yazdıralım.

	initialize global (degiskenim) to				
	Whe	n button1 💙 . Mouse Click 💙			
		switch get degiskenim			
		Case			
		do			
		Gase			
	Do	Set text1 V. Text V to ki			
		do Break			
_		Case			
		do Set text1 V. Text V to Uc "Uc "			
		Break			
		Şekil 49			

Sebit Eğitim ve Bilgi Teknolojileri A.Ş. 2020. Her Hakkı Saklıdır.

Şekil 48



- 1- Değeri 3 olan *degiskenim* adlı bir değişken tanımladık.
- 2- Butona tıkladığımız zaman ekrana yazmasını istediğimiz için Mouse Click bloğunu ekledik.
- 3- Değişkenimizi kontrol edebilmek için switch case bloğu ekledik ve çarka tıklayarak 2 tane daha durum yarattık.
- 4- Switch bölümüne değişkenimizi yerleştirerek kontrol edilebilmesini sağladık.
- 5- Case bölümlerine de olabilecek ihtimalleri yerleştirdik ve ekrana yazdırdık.

	Buton	
Üç		Çıktıda da görüldüğü gibi, değişkenimize 3 değerini verdik ve switch case karar kontrol yapısıyla kontrol ettirdikten sonra çıkan sonucu ekrana yazdırdık.
	Şekil 50	

# Döngüler (Loops)

Döngüler bir kod bloğunun birden çok kez tekrar edilmesini sağlayan yapılara verilen addır. Daha hızlı ve daha kullanışlı kod blokları oluşturmamızı sağlar. Örneğin, 1'den 5'e kadar olan sayıları ekrana yazdırmak istediğimizde bu sayıları tek tek yazmamız gerekirken döngüleri kullanarak bu sayıları tek seferde ekrana yazdırabiliriz. VFabrika'da bulunan döngü türleri şunlardır; Foreach (Her Eleman İçin), While (Sürekli Yap) ve Do While (Yap ve Doğruysa Sürekli Yap) döngüsü.

#### Foreach Döngüsü (Her Eleman İçin Döngüsü)

Bu döngüde koşul sağlanana dek bloklar çalıştırılır. Üç temel özelliği vardır; from (başlangıç), to (bitiş) ve by (artış). From kısmında döngünün hangi değerden başlayacağını, to kısmında şartı, by kısmında da başlangıç değerinin her döngüde kaç arttırılacağını belirtiriz.





Şekil 51: Foreach

Örneğin, varsayılan olarak gelen blokta from 0, to 10, by 1'dir. Buna göre, döngü 0'dan başlar, değerin 10'dan büyük olup olmadığını kontrol eder, büyük değilse 1 arttırarak do kısmındaki blokları çalıştırır. Değer 10'dan büyük olduğunda şart sağlanmış olur ve döngüden çıkılır.

*index* isimli varsayılan olarak gelen değer bizim bu döngüdeki değişkenimizdir. Bu değişken local (yerel) bir değişken olmakla birlikte do kısmında bu değişkenimizi kullanabiliriz. Foreach döngüsü *index'i* döngü içerisinde arttırabilir fakat başka bir değişken kullandığımızda eğer biz arttırmadıysak döngü tarafından arttırılmaz.

# While Döngüsü (Sürekli Yap Döngüsü)



Şekil 52: While

Foreach döngüsü gibi bu döngü de bir şarta bağlı olarak tekrarlama işlemlerini yapar. While kısmında belirtilen şart, do kısmında ise şart sağlandığı sürece yapılacak olan bloklar yer alır.

Döngüyü bitirecek bir şart olmazsa döngü sürekli olarak kendini tekrar eder ve sonsuz bir döngüye girebilir.

### Do While Döngüsü (Yap ve Doğruysa Sürekli Yap Döngüsü)

Foreach ve While döngülerinde önce şart kontrol edilir, eğer şart sağlanıyorsa döngü içindeki bloklar çalıştırılırdı. Do while döngüsünün en önemli farkı da do kısmında belirtilen blokların ilk olarak şart kontrol edilmeden uygulanmasıdır. Daha sonra while kısmındaki şart kontrol edilerek tekrarlama işlemi devam eder ya da döngü biter.



Şekil 53: Do While





## Break (Durdur)



Sekil 54: Break

Break bloğuna Karar Kontrol Yapılarında Switch Case başlığında değinmiştik. Orada olduğu gibi döngülerde de aynı şekilde kullanılmaktadır. Belirtilen bir şart sağlandığında döngüden çıkılmasını istiyorsak Break bloğunu kullanırız.

## Continue (Devam Et)



Şekil 55: Continue

Continue bloğu kullanıldığında döngüden çıkılmaz fakat döngüde bir sonraki tekrara geçilir. Örneğin, 1'den 10'a kadar olan sayıları yazan bir programda "4'e eşit olduğunda" gibi bir şart koyup Devam Et bloğunu eklersek 4 atlanarak direkt 5'e geçilir. Continue bloğundan sonraki bloklar atlanarak döngü 1 atlanır.

Uygulama: Foreach döngüsünü kullanarak 0'dan 11'e kadar olan sayıları konsola yazdıralım.





- 2- Foreach döngüsünde *index* bizim local değişkenimizdir. Bu sebeple *index* isimli değişkenimiz 0'dan başlar, 10'a kadar 1 artarak devam eder fakat 10 da işleme dâhil edilir.
- 3- Local değişkenimizi konsola yazdırmak istediğimiz için Console.WriteLine bloğunu kullandık ve ona da get erişimcisi ile *index* isimli değişkenimizi yerleştirdik.

Sebit Eğitim ve Bilgi Teknolojileri A.Ş. 2020. Her Hakkı Saklıdır.



🕞 💼 📔 Elements	Console » 🗛 1 🗄 🗙	
🕩 🛇 top	▼ Filter Default levels ▼	
▲ The SSL certificate used <u>dataLevel.html:1</u> to load resources from <u>https://appjsframewo</u> <u>rk.sebitvcloud.com</u> will be distrusted in M66. Once distrusted, users will be prevented from loading these resources. See <u>https://g.co/chrome/symantecpkicerts</u> for more information.		
0	<u>vfabrika-plugins.js:3784</u>	
1	<u>vfabrika-plugins.js:3784</u>	
2	<u>vfabrika-plugins.js:3784</u>	
3	<u>vfabrika-plugins.js:3784</u>	
4	<u>vfabrika-plugins.js:3784</u>	
5	<u>vfabrika-plugins.js:3784</u>	
6	<u>vfabrika-plugins.js:3784</u>	
7	<u>vfabrika-plugins.js:3784</u>	
8	<u>vfabrika-plugins.js:3784</u>	
9	<u>vfabrika-plugins.js:3784</u>	
10	<u>vfabrika-plugins.js:3784</u>	
>	Şekil 57	

Çıktıda da görüleceği üzere O'dan 11'e kadar olan sayıları konsola yazdırdık.

**NOT:** Konsol, metin tabanlı arayüz programıdır. Konsolda grafikler yer almaz. Birçok programlama dili öğretilirken konsol üzerinden öğretilir. Bunun sebebi de önce programlama ve algoritmanın öğretilmesidir. Ayrıca, konsollar kodları daha hızlı şekilde derleyebilir. VFabrika'da uygulamalarımızı yaparken varsayılan tarayıcıyı kullanırız. Her tarayıcı kendi konsoluna sahiptir ve konsol üzerinden de oluşturduğumuz blokları çalıştırabiliriz. Consol.WriteLine bloğu da konsola yazdırmamızı sağlar.

**Uygulama:** Bu uygulamamızda da değeri sıfır (0) olan bir değişken tanımlayıp while döngüsü ile 5'e kadar konsola yazdıralım.



Sayfa **| 34** 



Şekil 58

- 1- Değeri 0 olan *degiskenim* isimli global bir değişken tanımladık.
- 2- Butona tıkladığımız zaman ekrana yazmasını istediğimiz için Mouse Click bloğunu ekledik.
- 3- While döngüsünde get erişimcisi ile değişkenimizin değerini karşılaştırma operatörünü kullanarak 5'ten büyük olup olmadığını kontrol ettirdik.
- 4- Şartı sağladığı sürece konsola değişkenimizi yazdırdık.
- 5- Yine şartı sağladığı sürece; Get erişimcisi değişkenimizin değerini aldık, Plus one bloğu ile değişkenimizi 1 arttırdık. Değişkenimizin değerini değiştirmek istediğimiz için set erişimcisini kullandık.



🕞 💼 🛛 Elements	Console » 🔺 1 🚦 🗙	
🕩 🛇 top	▼ Filter Default levels ▼	
▲ The SSL certificate used <u>datalevel.html:1</u> to load resources from <u>https://appjsframewo</u> <u>rk.sebitvcloud.com</u> will be distrusted in M66. Once distrusted, users will be prevented from loading these resources. See <u>https://g.co/chrome/symantecpkicerts</u> for more information.		
0	<u>vfabrika-plugins.js:3784</u>	
1	<u>vfabrika-plugins.js:3784</u>	
2	<u>vfabrika-plugins.js:3784</u>	
3	<u>vfabrika-plugins.js:3784</u>	
4	<u>vfabrika-plugins.js:3784</u>	
>		

Şekil 59

Çıktıda da görüldüğü gibi, O'dan 5'e kadar olan sayılar konsola yazdırıldı. Blok sırasında önce değeri yazıp ardından arttırma işlemi yaptığımız için O'dan başlayıp 5'e kadar yazdırdı fakat 5'e geldiğinde şart (5'ten küçük olduğu sürece) sağlanmadığı için 5'i yazdırmadan döngüden çıktı.

#### Listeler

Şu ana kadar gördüğümüz değişkenlerde her zaman tek bir değeri tuttuk. Listeler bizim aynı değişkende birden çok değer tutmamızı sağlayan yapılardır. Değişkenleri boş bir kutu olarak düşündüğümüzde; listeler, bölmeleri olan bir kutu olarak düşünülebilir. Örneğin; bir kişinin adını, soyadını ve oturduğu şehri listeler yardımıyla tek bir değişkende tutabiliriz. Listelerin en önemli özelliklerinden biri sıralı olmasıdır. Biz de listenin içindeki değerlere ulaşmak için bu sırayı kullanırız. Listelerin sıralı olmasındaki bir diğer önemli nokta da listelerde sıranın O'dan başlamasıdır. Bizim listedeki ilk değerimiz sıfırıncı index'te bulunur.

#### Create List (Liste Oluşturma)



Bir liste oluşturmak için Create List bloğunu kullanırız. Çarka tıklayarak istediğimiz kadar item (öge) ekleyebiliriz.

Şekil 60: Create

Liste oluştururken dikkat edilmesi gereken nokta, değişken kapsama alanıdır. Listeler local (yerel) değişkenlerde tanımlanamaz. Yalnızca

global (genel) değişkenlerde tanımlanabilir.


initialize global (listem) to Create List item 1 item 2 item 3

Şekil 61

**Uygulama:** 1, 2 ve 3 değerlerine sahip global bir liste oluşturalım.

Liste oluşturmak da değişken oluşturmakla benzerdir. Listede birden çok değer tutabildiğimiz için, çarka tıklayarak ekleyeceğimiz değer kadar item ekleyebiliriz.

### Create From Comma Text (Virgülle Ayrılmış Metinden Liste Oluştur)





Metin içeriğine sahip ögeleri daha kolay şekilde eklememizi sağlar. Liste oluştururken eklemek istediğimiz verileri tek tek bloğa eklemek durumunda kalırken, bu blok ile beraber daha kısa zamanda ögeleri ekleyebiliriz. Ekleyeceğimiz verileri virgül ile ayırmamız yeterlidir.

## Add Item (Öge Ekleme)



Şekil 63: Add Item

Listeye ekleyeceğimiz ögeleri her zaman en başında tanımlamak zorunda değiliz. Listelerde add item bloklar içinde de öge ekleyebiliriz. List bölümüne hangi listeye eklemek istediğimizi, item kısmına da eklemek istediğimiz ögeyi yerleştirebiliriz. Eklenen öge listede en son sıraya eklenir.

## Add Item Into (Sıraya Öge Ekle)



Şekil 64: Add Item Into

Listeye öge eklediğimizde listenin son sırasına eklenir fakat Add Item Into bloğu ile istediğimiz sıraya istediğimiz ögeyi ekleyebiliriz. Örneğin başlangıç pozisyonuna öge eklemek istersek Into bölümünde 0 değerini koyarız ve ögeyi ekleriz. Böylelikle var olan ögelerin de sırası değişir. 1.



sırada olan öge 2. sıraya, 2. sırada olan öge de 3. sıraya gelir.

### Remove Item (Öge Çıkarma)



Şekil 65: Remove Item

Listelere bloklar içinde öge ekleyebildiğimiz gibi öge de çıkarabiliriz. Bunun için Remove Item bloğu kullanılır. List bölümüne hangi listeden çıkarmak istediğimi, item kısmında da çıkarmak istediğimiz ögeyi yerleştirebiliriz.

### Remove Item At (Pozisyondaki Ögeyi Sil)



Şekil 66: Remove Item At

İstediğimiz ögeyi çıkarabildiğimiz gibi, index'ine göre de ögeleri listeden çıkarabiliriz. Bunun için Remove Item At bloğu kullanılır. List kısmına hangi listeden çıkarılacağı, index bölümüne de hangi sıradaki ögenin çıkarılacağı eklenir.

### Length (Liste Uzunluğu)



Length bloğu, listedeki öge sayısına ulaşmamızı sağlar. Tam sayı döndürür.

Şekil 67: Length

### Contains (Mevcut)



Program içinde hangi ögenin bulunup bulunmadığını kontrol eden Contains (Mevcut) bloğu, yalnızca true (doğru) ya da false (yanlış) değerlerini gösterir.

Şekil 68: Contains



### Get Item At (Poziyondaki Ögeyi Oku)



Get Item At bloğu, girilen index sayısına bağlı olarak listede o index'teki değeri getiren bloklardır.

Şekil 69: Get Item At

### Index Of (Pozisyon Bul)



Listenin içinde pozisyonunu bulmak istediğimiz değerler olduğunda Index Of bloğunu kullanırız. List kısmında listeyi, Index Of'a da değeri yerleştiririz. Bu işlemin ardından VFabrika bize ögenin pozisyonu döndürür.

### Set Item At (Pozisyondaki Ögeyi Değiştir)



gerekebilir. Bu tür durumlarda Set Item At bloğunu kullanırız. From kısmında hangi listeye bakacağını, At kısmında index'i ve Value kısmında da yeni değeri yerleştirebiliriz. Bu blok yardımıyla, örneğin, 3. index'teki ögenin değerini kolaylıkla değiştirebiliriz.

Program içinde listedeki değerleri değiştirmemiz

Şekil 71: Set Item At

### Shuffle List (Listeyi Karıştır)



Şekil 72: Shuffle List

Listelerde ögeler her zaman sıralı hâlde bulunur fakat bu sırayı değiştirmek gerekebilir. Shuffle List, listedeki pozisyon sırasını kolayca değiştirmemizi sağlayan blok tipidir. Liste oluşturulurken veya blok yapıları içinde kullanılabilir.

### Sort (Sırala)



Listelerdeki ögeleri alfabetik olarak baştan sona ya da sondan başa sıralamamızı sağlayan bloktur.

**Uygulama**: *listem* isimli bir liste oluşturalım ve 0'dan 5'e kadar olan tam sayıları foreach döngüsü kullanarak listeye ekleyelim ve ekrana yazdıralım.



Şekil 70: Index Of

initia	lize global (listem) to K 🔅 Create List
Whe	n button1 . Mouse Click
Do	do do item get index
	Set text1 . Text to To Text get listem

Şekil 74

- 1- Global bir *listem* isimli liste oluşturduk.
- 2- Butona tıkladığımız zaman ekrana yazmasını istediğimiz için Mouse Click bloğunu ekledik.
- 3- Foreach döngüsü ekledik O'dan 5'e kadar 1 artacak şekilde düzenledik.
- 4- Add Item List ile oluşturduğumuz listeye 0'dan 5'e kadar olan değerleri ekledik. Index değişkeni yalnızca foreach döngüsü içinde kullanıldığından dolayı da onu yerleştirdik.
- 5- Set Text ile de listemizi ekrandaki Metin kutusuna yazdırdık.

Buton	Çıktı da görüldüğü üzere, butona tıkladığımızda
0,1,2,3,4,5	listemize ögeleri ekledik ve ekrana da yazdırdık.

Şekil 75



## Prosedürler

### Prosedür Nedir?

Her zaman az bir kod bloğu ile çalışmayabiliriz. Böyle bir durumda aynı blokları tekrar ve tekrar yerleştirmek hem programı yorar hem de kullanıcının zamanını çalar. Prosedürler, kod bloklarını küçük parçalara bölmemizi ve daha kolay şekilde yönetmemizi sağlayan yapılardır. Programlamanın temel yapı taşlarından biri olan prosedürler, programın çeşitli yerlerinde tekrar tekrar kullanılabilir ve değiştirilebilir.

### Procedure (Prosedür Oluşturma)



Prosedür oluşturmak için Procedure bloğunu kullanırız. Prosedüre isim verebilir ve Do bölümüne de prosedürün yapmasını istediğimiz kod bloklarını yerleştirebiliriz. Çarka tıkladığımızda prosedüre istediğimiz kadar input (girdi) girebiliriz.

Şekil 76: Procedure

### Call (Çağırma)



Prosedür oluşturduktan sonraki diğer adım, oluşturduğumuz prosedürü çağırmaktır. Oluşturulan prosedür çağrılmadığı takdirde programa herhangi bir etkisi olmaz. Blokta bulunan açılır pencere ile çağrılmasını istediğimiz prosedürü seçebiliriz. Bunun yanında, programın istenilen yerinde ve istenildiği kadar prosedür çağrılabilir.

Bu da prosedürlerin en büyük avantajlarından biridir.

**Uygulama:** Konsola "merhaba" yazdıran bir prosedür oluşturalım ve onun nasıl çağırıldığına bakalım.



Görüldüğü üzere Procedure bloğuna istediğimiz bloğu/blokları ekledikten sonra Call bloğunu da prosedürün çağrılacağı yere yerleştiriyoruz. Böylelikle bir işlemi her seferinde tekrar tekrar yapmak zorunda kalmıyoruz.

### Return Result (Sonuç Çağırma)



Şekil 79: Return Result

sonucu gönderebiliriz.

Prosedürün değer döndürmesini istediğimiz durumlarda Return Result bloğu kullanılır. Prosedür bloğunda genellikle en altta bulunan Return Result bloğu opsiyoneldir. Örneğin, programın belirli yerlerinde 2 farklı sayının toplamını istiyorsak burada prosedürün içinde sayıları toplayıp Return Result bloğu ile de

### Call for Result (Sonuç Döndüren Prosedür Çağırma)



Değer döndüren prosedürler oluşturduğumuzda, döndürülen değerin blok içinde kullanılması gerekebilir. Bu tür durumlarda Call for Result bloğunu kullanırız.

**Uygulama:** İki değişkenin prosedür içinde toplama işlemini yapalım ve elde edilen sonucu yazdıralım.





Şekil 81

- 1- *degisken1* ve *degisken2* diye iki farklı değişken tanımladık.
- 2- Butona basıldığında değişkenlerin ilk tanımladığımız değerleri ile prosedürden toplama işleminin sonucu çağırdık ve text1'e yazdırdık.
- 3- *degisken1* ve *degisken2*'nin değerlerini değiştirdik.
- 4- Prosedürden değişkenlerin yeni değerleri ile toplama işleminin sonucu çağırdık ve text2'e yazdırdık.





## String (Metin)

Değişkenler konusu anlatılırken String ile bir veri tipi olarak karşılaştık. Şimdi de bu konuyu daha ayrıntılı olarak inceleyeceğiz.

String, tek bir öge gibi davranan karakterler dizisidir. String ile karakterler üzerinde oynamalar yapabilir ve karakterleri saklayabiliriz.



String değerini VFabrika'da Value (Değer) bloğu ile birlikte gireriz. Bu bloğa harfler, semboller ve sayılar girilebilir fakat hepsi karakter olarak işlenir. Örneğin, bloğun içine 15 yazdığımızda sayısal değer olarak değil karakter olarak işlenir. Aritmetik işlemler yapılmaz.

### Compare Text (Karşılaştır)



Şekil 84: Compare

Girilen iki karakter dizisini karşılaştıran blok yapısıdır. Açılır pencerede; eşittir ve eşit değildir olmak üzere iki farklı seçenek vardır.

## Contains (İçeriyor mu)



Bir string içinde istenilen karakterin olup olmadığını kontrol eden blok yapısıdır. Text kısmında kontrol edilecek string, piece bölümünde ise aranacak karakter eklenir.

Şekil 85: Contains



### Is Empty (Boş mu)



Şekil 86: Is Empty

Eklenen string'in karakter içerip içermediğini kontrol eden blok yapısıdır. String karakter içeriyorsa True (Doğru), içermiyorsa False (Yanlış) değerini döndürür.

#### Join (Ekle)



Eklenen string'leri birleştirerek tek bir string hâline getiren blok yapısıdır. Üzerinde bulunan çarka tıklanarak istenildiği kadar bölme açılabilir. Birleştirme sırası yukardan başlayarak aşağıya doğru ilerler.

Şekil 87: Join

### Index of Text (Metnin Konumunu Bul)



Aranılan karakter veya karakterin string'in içinde kaçıncı pozisyonda olduğunu gösteren blok yapısıdır. Stringlerde pozisyon sırası O'dan başlar. Index of Text bloğu bulunan ilk pozisyonu döndürür. Örneğin, "mehmet" isminde "m" karakteri olup olmadığını kontrol ettirdiğimizde bize 0 değerini döndürecektir. Aranılan karakteri string içinde aramaya başlar ve ilk bulduğunda

Şekil 88: Index of Text

pozisyonu döndürür. String'in geri kalanına bakmaz. Aranan karakteri string içerisinde bulamazsa -1 değerini döndürür.

#### Length (Uzunluk)



String'deki karakter sayısını veren blok yapısıdır.

Şekil 89: Length

### Replace All (Hepsini Değiştir)



Şekil 90: Replace All

String içinde bulunan kelime veya cümleyi değiştirmeyi sağlayan blok yapısıdır. Text kısmına değiştirelecek olan string eklenir, segment kısmına hangi bölümün değiştirileceği belirtilir ve replacement bölümünde de hangi string ile değiştireğini ekleriz. Örneğin; "Ahmet kodlamayı seviyor." diye bir string'e sahip olalım. Segment kısmına "Ahmet" ekleyip replacement kısmına da "Ali" ismini eklersek yeni string'imiz "Ali kodlamayı seviyor." olacaktır.



### Segment (Böl)



Şekil 91: Segment

String'de belirtilen değerler arasındaki karakterleri almamızı sağlayan blok yapısıdır. Start ve Length kısımlarına başlangıç ve bitiş değerlerini ekleriz. Örneğin; "Programlama" diye bir string'imiz olsun. Start değerine 4 ve length değerine de 6 değerlerini eklersek çıktı olarak "ram" değerini alırız.

Aranılan karakter veya karakterin string'in içinde kaçıncı sıradan

başladığını gösteren blok yapısıdır. Stringlerde pozisyon sırası O'dan

başlar. Eğer tek bir karakter aranıyorsa o karakteri ilk bulduğu değeri

döndürür. Ancak, bir karakter topluluğu veya bir kelimeyi arıyorsa ilk

### Starts At (Başlangıç Pozisyonunu Bul)



Şekil 92: Starts At

### Trim (Boşlukları Temizle)



Trim bloğu string'in başında veya sonunda olan boşlukları kaldırır.

karakterlerinin başlangıç sırasını döndürür.

Şekil 93: Trim

### To Lowercase (Küçük Harfe Çevir)



Bu blok, sağ tarafına eklediğimiz metnin tüm karakterlerini küçük harfe çevirir.

Şekil 94: To Lowercase

### To Uppercase (Büyük Harfe Çevir)



Bu blok, sağ tarafına eklediğimiz metnin tüm karakterlerini büyük harfe çevirir.

Şekil 95: To Uppercase



### To Text (Metne Çevir)



Kimi durumlarda bazı değerleri metin değerine çevirmemiz gerekebilir. Örneğin, karşılaştırma operatörlerinde iki değişkeni karşılaştırıp eşit olup olmadığını bir metin kutusuna yazdırmak istediğimizde bunu yapamayız çünkü bu karşılaşmadan çıkan sonuç 1 (true) veya 0 (false) olur ve

program da bunu algılamayabilir. Bu tür durumlarda To Text bloğu kullanılır ve çıkan sonucun metin kutusuna metin olarak yazdırmamızı sağlar.

### Concat (Birleştir)



Concat, eklenen iki string'in tek bir string hâline çevrilmesini sağlayan blok yapısıdır.

### Split (Ayır)



Şekil 98: Split

Bir string'in istenildiği yerden parçalara ayrılmasını sağlayan blok yapısıdır. String ayrıldıktan sonra bir liste gibi olur. Girilen karakter veya karakterler string'in her yerinden kaldırılır. Örneğin; "Vitamin" kelimesinden "i" karakterini çıkardığımızda elde edeceğimiz çıktı üç elemanlı bir liste olacaktır. Elemanları da "V", "tam", "n" olur.

Uygulama: Verilen iki string'i karşılaştıralım ve sonucu ekrana yazdıralım.

When button 1 . Mouse Click		
Do Set text . Text to To Text	compare text T Programlamay: seviyorum) ==	Programlama "



- 1- Butona tıkladığımızda yazdırmasını istediğimiz için Mouse Click bloğunu ekledik.
- 2- "Programlamayı seviyorum" ve "Programlama" değerlerine sahip iki string'i Compare Text bloğuna ekledik.
- 3- Sonuç 1 veya 0 olarak çıkacağından sonucun metne çevrilmesi gerekiyor, bu sebeple To Text bloğunu kullandık.



	Buton
false	

Şekil 100



## **BÖLÜM 2 - FORM NESNELERİ**

İnternet sitelerini ziyaret ettiğimizde bizden çoğu zaman kullanıcı girişi yapmamızı ister. Bunun için de önümüze kullanıcı adımızı ve şifremizin yazılacağı iki kutu ve yazdıktan sonra göndereceğimiz bir buton bulunur.



Bu tür kullanıcı ile web sayfası arasındaki veri alışverişini sağlayan yapılar HTML'de **form** olarak adlandırılmaktadır. Formlar kullanıcıdan veri girişi almamızı sağlar. "Input (Girdi)" en önemli form etiketidir. Bu etiket JavaScript kodu yazarken kullanılmak zorunda fakat VFabrika'da hazır olarak form nesnelerine eklenebilmektedir. Input ile beraber birçok form nesnesini kullanabiliriz. Button (Buton), Image Button (Resim Butonu), Radio Button (Seçenek Butonu), Chechkbox (Kontrol Butonu), Dropdown List (Açılan Kutu), Textbox (Metin Kutusu), Slider (Kaydırıcı) VFabrika'da kullanılan form nesneleridir.

## **Button (Buton)**



Buton, sahneye tıklanabilir bir düğme eklememizi sağlar. Böylelikle oluşturduğumuz blok veya blok yapılarını tetikleyebiliriz. Dizayn bölümünde iken Input (Girdi) penceresinden sürükle bırak yöntemiyle sahneye buton ekleyebiliriz. Bloklar'a geçtiğimizde, bu nesneye özel blokları Button

penceresinde bulabiliriz.

When . Mouse Click On Mouse Event	On mouse event (Fare Olayı) bloğu ile Button'a tıklandığında hangi işlemleri yapacağımızı belirleriz.





Set Property (Özellik Ayarla): Sahneye eklediğimiz nesnelere tıkladığımızda, sağ altta Properties (Özellikler) isimli bir pencere görebiliriz. Bu pencerede tıkladığımız nesnenin özelliklerini değiştirebiliriz. Dizayn bölümünün dışında bloklar yardımı ile de nesnelerin özelliklerine ulaşabiliriz. Set Property (Özellik Ayarla), bloğun özelliğini dilediğimiz şekilde ayarlamamızı sağlayan kod bloğudur. Set kısmındaki ilk açılır pencereden istediğimiz bloğu seçip ikinci açılır pencereden de ayarlamak istediğimiz özelliği seçebiliriz.

Get Property (Özellik Oku): Nesnelerin özelliklerinin okunmasını sağlayan kod bloğudur. Örneğin, bir butonun yüksekliğini, font büyüklüğünü Get Property (Özellik Oku) yardımıyla rahatlıkla öğrenebiliriz.

Set Property (Özellik Ayarla) ve Get Property (Özellik Oku) blokları tüm form nesnelerinde yer alır ve hepsinde de o nesneye özel olarak bulunur. Örneğin; bloklar bölümünde, Button penceresindeki bir Set Property bloğunu eklediğimizde bu bloğu başka herhangi bir nesne için kullanamayız.

## Image Button (Resim Butonu)

Text Şekil 105 Resim butonu tıpkı button gibi sahneye tıklanabilir bir düğme eklememizi sağlayan form nesnesidir. Butondan farklı olarak standart bir görüntü yerine resim ekleyebilir ve öyle kullanabiliriz. Bunun için de sahnemize eklediğimiz resim butonuna tıklayarak özellikler penceresine gidip "Durum Resim

Url'leri" başlığından resmi seçip ekleyebiliriz.

Ayrıca, resim butonuna tek bir resim ekleyebildiğimiz gibi Sprite (Resim Grubu) da ekleyebiliriz.

When A Mouse Click On Mouse Event	On Mouse Event (Fare Olayı) ile resim butonuna tıklandığında yapılmasını istediğimiz blokları çalıştırabiliriz.
Image Button Text Align Top Left Text Align	Text Align (Metin Konumu) bloğu ile resim butonunda bulunan metnin pozisyonunu ayarlayabiliriz.



## Radio Button (Seçenek Butonu)

Birçok seçenek içinden tek bir seçeneğin seçilmesine imkân veren form nesnesidir. Dizayn bölümünde iken Input (Girdi) penceresinden sürükle bırak yöntemiyle sahneye şekil 106 istenildiği kadar radio buton ekleyebiliriz.



**NOT:** Radio button denmesinin sebebi, eski radyolardaki düğmelere benzeyişi ve işleyiş şeklinin aynı olmasıdır. Eski radyolarda da bir düğmeye basıldığında eğer basılı bir düğme varsa iptal olurdu ve yalnızca tek bir düğme basılı olarak kalırdı.

## **CheckBox (Kontrol Butonu)**



Seçenekler içinden birden fazla seçeneğin seçilmesine olanak sağlayan form nesnedir. Dizayn bölümünde iken Input (Girdi) penceresinden sürükle bırak yöntemiyle sahneye ekleyebiliriz.

Şekil 107

Radio Button ile en önemli farkı, radio button yalnızca tek bir seçeneğin seçilmesine imkân verirken checkbox ile birden fazla seçeneğin seçilebilmesidir.





## Dropdown List (Açılan Kutu)



Açılır kapanır bir menü oluşturmamızı ve içinde çeşitli seçenekler barındırmamızı sağlayan form nesnesidir. Dizayn bölümünde iken Input (Girdi) penceresinden sürükle bırak yöntemiyle sahneye ekleyebiliriz. Açılan kutunun içine öge eklemesini, dizayn bölümünden veya bloklar aracılığı ile yapabiliriz. Dizayn bölümünden açılır listeye tıkladığımızda özellik menüsü çıkacaktır. Orada "Nesneler" kısmındaki üç noktaya tıklayarak seçeneklerimizi

ekleyebiliriz. Bloklar ile de bunu yapabiliriz.



Öge ekleyebildiğimiz gibi var olan ögeleri de kaldırabiliriz. Bunu hem özellikler menüsünden hem de bloklar ile yapabiliriz.

Remove Item Block	Remove Item Block (Öge Çıkar Bloğu) en sondaki ögeyi siler.
Remove Item At Index 0 Remove Item At Block	Remove Item At Block (Ögeyi İndeksten Çıkar Bloğu) girdiğimiz index sayısına göre o sırada bulunan seçeneği siler.
Clear	Clear (Temizle) bloğu, seçilmiş olan bir öge varsa bu seçimi geri alır ve listeyi varsayılan hâline geri getirir.

Hangi seçeneğin seçildiğini öğrenebilmek için Radio Button ve CheckBox nesnelerinde de olduğu gibi Get Property (Özellik Ayarla) bloğunu kullanabiliriz. Bu blokta bulunan *Item, Text, Value, Index* değerlerinden birini seçerek ona göre de kontrolü sağlayabiliriz.



## **Textbox (Metin Kutusu)**

Şekil 109

Kullanıcıdan metin girdisi almamızı sağlayan form nesnesidir. Dizayn bölümünde iken Input (Girdi) penceresinden sürükle bırak yöntemiyle sahneye ekleyebiliriz. Metin Kutusuna özellikler menüsü ve bloklar aracılığı ile birtakım özellikler ekleyebiliriz.

When Text Changed Do On Text Changed Event	On Text Changed Event (Metin Değişme Olayı) ile metin değiştiği zaman neler yapılacağını belirleyebiliriz.
Set highlight color to None Set Highlight Color	Highlight (Vurgu) bloğu ile vurgu rengini değiştirebiliriz.
Check Validation Check Validation	Check Validation (Doğrulamayı Kontrol Et) bloğu metin kutusunun doldurulmasının gerekli olduğu durumlarda kullanılır.
Tooltip Position	Tooltip Position (İpucu pozisyonu) ile yazılan ipucunun metin kutusunun hangi tarafında olacağını belirtebiliriz.
Ignore Key	Ignore Key (Tuşu Yoksay) bloğu klavye ile veri girişini engeller.
When keypress Do On Keypress Event	On Keypress Event (Tuşa Basılma Olayı) bloğu metin kutusunda klavyeden bir tuşa basıldığında neler yapılacağını belirlememizi sağlar.





## Slider (Sürgü)



Slider nesnesi, sürgüyü kaydırdığımız zaman oluşturulan blok yapılarının çalışmasını sağlayan form nesnesidir. Dizayn bölümünde iken Input (Girdi) penceresinden sürükle bırak yöntemiyle sahneye ekleyebiliriz. Slider nesnesinde varsayılan olarak 100 bölme

bulunur. Bu da sürgüyü çok fazla kaydırabileceğimiz anlamında gelir. Bu sayıyı slider kullanma amacımıza göre azaltabilir veya arttırabiliriz. Dizayn bölümünde slider'a tıkladığımızda çıkan davranış özelliklerinden "En Büyük Değer", "En Küçük Değer", "Adım Büyüklüğü" değerleriyle değiştirebiliriz.

When Y. Mouse Down Y Do On Mouse Event	On Mouse Event (Fare Olayı) bloğu ile slider'a tıklandığında yapılmasını istediğimiz blokları çalıştırabiliriz.
When Value Changed On Value Change Event	On Value Change Event (Değer Değişme Olayı) bloğu ile de sürgü her kaydırıldığında, yani sürgünün değeri değiştiğinde, yapılmasını istediğimiz blokları çalıştırır. Slider'ın en büyük değerini, en küçük değerini ve adım büyüklüğünü davranış özelliklerinden değiştirebildiğimiz gibi Set Property bloğu ile de değiştirebiliriz.



# **BÖLÜM 3 – MEDYA NESNELERİ**

## Image (Resim)



Yaptığımız projenin görünümünü değiştirmek veya yeni görseller eklemek için resim nesnesini kullanabiliriz. Dizayn bölümünde iken Medya penceresinden sürükle bırak yöntemiyle sahneye resim ekleyebiliriz. Sahneye bıraktığımız anda bizden eklemek istediğimiz resmin adresini isteyecektir, resmin konumunu seçtikten sonra resmimiz sahneye eklenir.

Şekil 111

Ayrıca, resim nesnesinde Sprite Image (Resim Grubu) özelliğini kullanabiliriz. Sprite Image ile basit gif animasyonları hazırlanabilir.

When A Mouse Click On Mouse Event	On Mouse Event (Fare Olayı) bloğu ile eklenilen resme tıklandığında hangi işlemleri yapacağımızı belirleriz.
When Con Drag Event	On Drag Event (Sürükle) bloğu ile resmi sürüklediğimizde yapacağımız işlemleri belirleriz.
When Drop Do On Drag and Drop Event	On Drag and Drop Event (Sürükle Bırak) bloğu ile de resmi bıraktığımızda yapılacak işlemleri belirleyebiliriz.

**NOT:** Bir resmi veya bir nesneyi sürükle bırak yapabilmek için, o nesne ile bırakılacağı yerin aynı Content (İçerik) içinde olması gerekmektedir. Content (İçerik) nesnesi Bölüm 4'te detaylı şekilde anlatılacaktır. Bunun yanında, resmin sürüklenebilir ve bırakılabilir özelliklerinin açık olması gerekmektedir. Özellikler penceresinden veya Set Property (Özellik Ayarla) bloğu ile bunları yapabiliriz.

When On Animation Control Event	On Animation Control Event (Animasyon Olayı) bloğu ile oluşturduğumuz resim animasyonlarını kontrol edebiliriz. Örneğin, animasyon başladığında, bittiğinde, durduğunda veya frame'i
---------------------------------	---



	değiştiğinde neler vapılacağı belirtilebilir
	yapılacağı belli tilebilli .
Animation Play Animation	Oluşturulan animasyonu oynatmayı veya durdurmayı sağlayan blok yapısıdır.
Animation Go To And Play Animation GoTo	Animation GoTo (Animasyon Karesi Ayarla) bloğu oluşturulan animasyonun frame'ini değiştirip aynı zamanda oynatma veya durdurma işlemlerini yapar.

## Sound (Ses)

Şekil 112

Projeye ses eklememizi sağlayan medya nesnesidir. Eklediğimiz sesleri kolaylıkla kontrol edebiliriz. Dizayn bölümünde iken Medya penceresinden sürükle bırak yöntemiyle sahneye eklediğimizde karşımıza çıkan pencereden ses dosyasının konumunu girerek ekleyebiliriz.

When On Player Control Event	On Player Control Event (Oynatıcı Kontrolü Olayı) bloğu ile ses başladığında, durduğunda veya bittiğinde neler yapılacağını belirtebiliriz.
Sound Play Player Control	Eklediğimiz ses nesnesinin oynatılmasını, durdurulmasını veya duraklatılmasını Player Control (Oynatıcı Kontrolü) bloğu ile yapabiliriz.

### Video



Şekil 113

Video nesnesi ile projemize videolar ekleyebiliriz. Video nesnesini sahneye eklediğimizde bizden poster ve video adresinin konumunu isteyen bir pencere çıkar, bu pencereden dosyamızın konumunu verebilir ve aynı zamanda videonun özelliklerinden olan "Tekrarla", "Kontroller" ve "Otomatik Oynat" seçeneklerini de seçebiliriz.



When Control Event	On Player Control Event (Oynatıcı Kontrol Olayı) bloğu video nesnesini birçok şekilde kontrol etmemizi sağlayan blok yapısıdır. Video başladığında, durduğunda, bittiğinde, yüklendikten sonra ve oynarken hangi blokların çalıştırılacağını belirtebiliriz.
Video Play Player Control	Eklediğimiz video nesnesinin oynatılmasını, durdurulmasını veya baştan başlatılmasını Player Control (Oynatıcı Kontrolü) bloğu ile yapabiliriz.



Player GoTo (Oynatıcıyı Götür) bloğu, istenilen saniyeye gidip videoyu oynatmayı veya durdurmayı sağlar.



# **BÖLÜM 4 – TASARIM NESNELERİ**

## Canvas (Resim Kâğıdı)



Canvas (Resim Kâğıdı) sahnemize çizim yapmamızı sağlayan tasarım nesnesidir. Canvas nesnesinde çizimler bloklar yardımı ile yapılır. Dolayısıyla, sahneye eklenen canvas nesnesi bloklar ile yaptığımız çizimleri tutar.

Şekil 114

Canvas Context	Context (Ortam) Çizim yapılacak canvas nesnesini seçmemizi ve aynı zamanda eklenen çizim bloklarının da döndürülmesini sağlayan blok yapısıdır.
Begin Path Close Path Close Path	Begin Path (Yola Başla) ve Close Path (Yolu Bitir) blokları çizimin başlangıç ve bitiş yolunu belirtmemizi sağlar.
Move To X Y O Move To	Move To (Taşı) bloğu çizimin nereden başlayacağını belirtir fakat herhangi bir çizim oluşturmaz.
Line To X Y	Line To (Çizgi Çiz) bloğu noktaları belirterek çizgi çizmemizi sağlar.
Rect X O Y O Width O Height	Rect
Rect (Kare) bloğu girdiğimiz değerlere göre bir görünmez.	kare oluşturmamızı sağlar fakat sahnede
Stroke Rect X (0) Y (0) Width (0) H	leight O Stroke Rect
Stroke Rect (Boş Kare Çiz) içi boş bir kare çizme	mizi sağlar.

Fill Rect X (0) Y (0) Width (0) Height (0)	Fill Rect

Fill Rect (Dolu Kare Çiz) kare oluştururken içinin de dolu olmasını istediğimiz durumlarda kullanılabilir.



Clear Rect (Alanı Temizle) bloğu ile girilen koordinat, yükseklik ve genişliğe göre alanı temizler.

Stroke	Stroke(Çizgi) bloğu ile oluşturduğumuz karede eğer çizgiler yoksa çizgi eklememizi sağlar. Örneğin; Rect (Kare) bloğu eklediğimizde sahnede bir kare görünmediğinden bahsetmiştik, stroke bloğu ile rect bloğuna çizgi ekleyebiliriz.
Fill	Fill (Doldur) bloğu ile oluşturulan karelerin içini doldurabiliriz.
Stroke Style #000000 Stroke Style	Stroke Style (Çizgi Stili) ve Fill Style (Dolgu Stili) blokları ile çizgi ve dolgulara renk verebiliriz. Renkleri kelime (red, green, black gibi) ile ya da RGB renk kodları ile belirtebiliriz.
Line Width	Line Width (Çizgi Kalınlığı) bloğu ile çizgilerin kalınlığını değiştirebiliriz.
Line Cap Square Line Cap	Line Cap (Çizgi Ucu) bloğu çizilen çizgilerin ucunun nasıl olacağını belirler. Kare, yuvarlak ve kapaksız şeklinde üç farklı dizayn verebiliriz.



Sayfa | 59







## **Content (İçerik)**

		11

Content (İçerik), sahneye eklenen nesneleri gruplandırmamızı sağlar. Dolayısıyla bir content nesnesinin özelliğini değiştirdiğimizde, örneğin gizlediğimizde, içindeki tüm nesnelere uygulamış oluruz. Buna ek olarak, Drag-Drop (Sürükle Bırak) yapılacak nesnelerin de aynı content içinde bulunmaları gerekmekte. Eğer drag

Şekil 115

yapılacak obje bir content'in içinde ise drop edileceği yer de aynı content'te olmalı.

When . Drop Do On Drag Drop Event	On Drag Drop Event (Sürükle Bırak Olayı) bloğu ile nesne bırakıldığında neler yapılacağını belirleyebiliriz.
When Con Drag Event	On Drag Event (Sürükle Olayı) bloğu ile nesne sürüklenmeye başladığında neler yapılacağını belirleyebiliriz. Sürüklenme sürecinde dört farklı seçenek vardır. Bunlar; Drag Begin (Sürüklemeye Başladığında), Drag Move (Sürükleme Hareketinde), Drag Ending (Sürükleme Biterken), Drag End (Sürükleme Bittiğinde).
Get Drag Object Get Drag Object	Get Drag Object (Sürüklenen Nesneyi Oku) bloğu sürüklenen nesnenin tüm özelliklerini (adını, pozisyonunu, görünürlüğünü vs.) öğrenmemizi sağlar.
Get Drag Object Name Get Drag Object Name	Get Drag Object Name (Sürüklenen Nesnenin Adını Oku) bloğu Get Drag Object'ten farklı olarak sürüklenen nesnenin yalnızca adını öğrenmemizi sağlar.
Get Drop Object Get Drop Object	Get Drop Object (Bırakılan Nesneyi Oku) bloğu, bırakılan nesnenin tüm özelliklerini (adını, pozisyonunu, görünürlüğünü vs.) öğrenmemizi sağlar.
Get Drop Object Name Get Drop Object Name	Get Drop Object Name (Bırakılan Nesnenin Adını Oku) bloğu, Get Drop Object'ten farklı olarak bırakılan nesnenin yalnızca adını öğrenmemizi sağlar.



## Table (Tablo)



Tablo nesnesi, sahnemize yeni tablolar eklememizi ve içinde de istediğimiz verilerin listelenmesini sağlar. Tabloyu dilediğimiz gibi biçimlendirebiliriz, böylelikle tasarımını da belirleyebiliriz. Dizayn bölümünde bulunan özellikler penceresinden tabloya sütun ekleyebilir, satır yüksekliklerini ayarlayabilir ve diğer özelliklerini değiştirebiliriz.

Şekil 116

Add Row Add Row	Add Row (Satır Ekle) bloğu ile tabloya bir satır ekleyebiliriz.
Add Row At Add Row At	Add Row At (Pozisyona Satır Ekle) bloğu, tabloda girilen konuma satır ekler.
Remove Row Remove Row	Remove Row (Satır Sil) bloğu ile tabloda en altta bulunan satırı silebiliriz.
Remove Row At Remove Row At	Remove Row At (Pozisyondaki Satırı Sil) bloğu, tabloda istenilen pozisyondaki satırı siler.
Clear Clear	Clear (Temizle) bloğu; tablodaki satır, sütun ve verileri siler.
Set Column by Index	et Column By Index
Set Column By Index (Pozisyondaki Sütun Metnini Ay değiştirmemizi, metin yoksa da eklememizi sağlar. Ta	arla) bloğu, girilen pozisyondaki metini Iblodaki son satıra ekler.
Set Column by Name	Set Column By Name
Set Column By Name (Sütun Metnini Ayarla) bloğu da	a sütunun ismini girerek veri eklememizi

veya değiştirmemizi sağlar. Tablodaki son satıra ekler.

Set Column At Row Row O Index Set Column At Row
Set Column At Row (Pozisyondaki Sütun Metnini Ayarla) bloğu ile girilen satır ve pozisyona istenilen veriyi eklememizi sağlar.

NOT: Satırların (Row) pozisyonu 1'den başlıyor fakat sütunların (Column) pozisyonu 0'dan

başlıyor.

Scroll To Bottom Row	Scroll To Bottom Row (Son Satıra Kaydır) bloğu, tablonun aşağıdaki satırlarına gitmemizi sağlar.
Set Footer Text	Set Footer Text (Alt Başlığı Ayarla) bloğu, alt başlığının metnini değiştirmemizi sağlar.
Set Selected Cell Column Index Set Selected Cell Row Index	Set Selected Cell (Seçili Hücreyi Ayarla) bloğu ile satır ve sütunu girilen hücreyi işaretleyebiliriz.
Get Selected Column Index Get Selected Column Index	Get Selected Column Index (Seçili Sütun Pozisyonu) bloğu, seçili olan bir hücre varsa onun sütununu öğrenmemizi sağlar.
Get Selected Row Index Get Selected Row Index	Get Selected Row Index (Seçili Satır Pozisyonu) bloğu, seçili olan bir hücre varsa onun satır bilgisini öğrenmemizi sağlar.
Clear Selection	Clear Selection (Seçimi Temizle) bloğu, seçili olan bir hücre varsa ondaki seçimi kaldırmamızı sağlar.



## Highlight (Vurgu)



Hightlight (Vurgu) nesnesi, sahnede dikkat çekilmesini istediğimiz yer veya nesnelerde kullanılabilir. Örneğin, metin kutusuna veri girilmesini istediğimizde vurgu ekleyerek kullanıcının dikkatini metin kutusuna çekebiliriz. VFabrika'da birçok farklı vurgu türü vardır; dikkat çekilmesi istenen nesneye göre vurgu türünü seçebiliriz. Vurgu türlerine dizayn bölümündeki özellikler penceresinden

ulaşabiliriz.

Şekil 117

Show Show	Show (Göster) bloğu, vurgu nesnesinin sahnede gösterilmesi sağlar.
Hide	Hide (Gizle) bloğu, vurgu nesnesinin sahnede gizlenmesini ve gösterilmemesini sağlar.
Hide All Highlights Hide All Highlights	Hide All Highlights (Tüm Vurguları Gizle) bloğu ile Hide (Gizle) nesnesinden farklı olarak sahnedeki tüm vurgu nesnelerini gizleyebiliriz.

## **Primitive Objects (Temel Nesneler)**



Primitive Objects (Temel Nesneler) sahneye geometrik şekiller eklememizi sağlar. Bu şekiller; elips, dikdörtgen, yuvarlatılmış diktörtgen ve üçgendir. Temel nesnelerdeki geometrik şekiller statiklerdir.

Şekil 118

When A Mouse Click On Mouse Event	On Mouse Event (Fare Olayı) bloğu ile temel nesnelere tıklandığında neler yapılacağını belirleyebiliriz.
Get Color Get Fill Color	Get Fill Color (Dolgu Rengini Oku) bloğu, nesnelerin içini dolduran dolgu renklerini okumamızı sağlar.
Set Fill Color to Set Fill Color	Set Fill Color (Dolgu Rengini Ayarla) bloğu, dolguların rengini seçmemizi sağlar.
Get Color Get Stroke Color	Get Stroke Color (Çizgi Rengini Oku) bloğu, temel nesnelerin çizgilerinin hangi renk olduğunu öğrenmemizi sağlar.
Set Stroke Color to	Set Stroke Color (Çizgi Rengini Ayarla) bloğu, temel nesnelerin çizgi rengini düzenler.



Get Get Stroke Thickness	Get Stroke Thickness (Çizgi Kalınlığını Oku) ile temel nesnelerin sahip oldukları çizgi kalınlıklarını öğrenebiliriz.
Set Stroke Thickness to Set Stroke Thickness	Set Stroke Thickness (Çizgi Kalınlığını Ayarla) bloğu ile temel nesnelerin çizgi kalınlıklarını istediğimiz değerde yapabiliriz.
Get Corner Radius Get Corner Radius	Get Corner Radius (Köşe Yarıçapını Oku) bloğu ile temel geometrik şekillerin köşe yarıçaplarını öğrenebiliriz.
Set Corner Radius to Set Corner Radius	Set Corner Radius (Köşe Yarıçapını Ayarla) bloğu ile temel geometrik şekillerin köşe yarıçaplarını ayarlayabiliriz.

## Smart Objects (Akıllı Nesneler)

VFabrika'da iki tür akıllı nesne vardır: Parallax Scroll (Paralaks Kaydırma) ve Polygon (Çokgen).

### Parallax Scroll (Paralaks Kaydırma)



Parallax Scroll (Paralaks Kaydırma) belirli bir hızda akan resimler oluşturmamızı sağlayan akıllı nesnedir. Örneğin; sürekli hareket eden bir yol ve etrafındaki ağaçları paralaks vasıtasıyla kolaylıkla yapabiliriz. Bunun için, yol resmini ve ağaç resmini paralaksın içine atıp başlatmamız yeterlidir.

Şekil 119

Sahnemizdeki paralaks nesnesine tıkladığımızda özellikler penceresinde "Kayan Görseller" yazısının yanında bulunan üç noktaya tıkladığımızda kaymasını istediğimiz resimleri ekleyebiliriz.

Play	Play (Oynat) bloğu ile paralaksı başlatabiliriz.
Stop Stop	Stop (Durdur) bloğu ile kayan paralaks resimlerini durdurabiliriz.
Set Speed Set Speed	Set Speed (Hızı Ayarla) bloğu ile paralaksın hızını ayarlayabiliriz.
Set Direction	Set Direction (Yönü Ayarla) bloğu ile paralaksın kayacağı yönü seçebiliriz. Paralaks nesnesi varsayılan olarak Sol tarafa doğru kayar.



When A Mouse Click On Mouse E	On Mouse Event (Fare Olayı) bloğunda, paralaks nesnesine tıklandığında neler yapılacağını belirleyebiliriz.

## Polygon (Çokgen)



Polygon (Çokgen), sahneye geometrik şekiller eklememizi sağlayan akıllı nesnedir. Primitive Objects'in (Temel Nesneler) statik olduğu bir önceki başlıkta belirtilmişti. Çokgenler dinamiktir. Dolayısıyla köşe sayıları ve başlangıç açıları değiştirilebilir.

Şekil 120

Get Count Get Edge Count	Get Edge Count (Köşe Sayısını Oku) bloğu, çokgenin kaç köşeye sahip olduğunu gösterir.
Set Edge Count to	Set Edge Count (Köşe Sayısını Ayarla) bloğu, çokgenin köşe sayısını değiştirmemizi sağlar. Böylelikle üçgen, beşgen, yedigen gibi geometrik şekiller oluşturabiliriz.
Get Cet Starting Angle	Get Starting Angle (Başlangıç Açısını Oku) bloğu ile çokgenin başlangıç açısını öğrenebiliriz.
Set Start Angle to Set Start Angle	Set Start Angle (Başlangıç Açısını Ayarla) bloğu ile çokgenin başlangıç açısını değiştirebiliriz.
Get Color Get Fill Color	Get Fill Color (Dolgu Rengini Oku) bloğu, çokgenlerin içini dolduran dolgu renklerini okumamızı sağlar.
Set Fill Color to	Set Fill Color (Dolgu Rengini Ayarla) bloğu, dolguların rengini seçmemizi sağlar.
Get Color Get Stroke Color	Get Stroke Color (Çizgi Rengini Oku) bloğu, çokgenlerin çizgi renklerini okur.
Set Stroke Color to Set Stroke Color	Set Stroke Color (Çizgi Rengini Ayarla) bloğu, çokgenlerin çizgi rengini düzenler.



Get Get Stroke Thickness Get Stroke Thickness	Get Stroke Thickness (Çizgi Kalınlığını Oku) ile çokgenlerin çizgi kalınlıklarını öğrenebiliriz.
Set Stroke Thickness to	Set Stroke Thickness (Çizgi Kalınlığını Ayarla) bloğu ile çokgenlerin çizgi kalınlıklarını ayarlayabiliriz.
When A Mouse Click On Mouse Event	On Mouse Event (Fare Olayı) bloğunda, çokgene tıklandığında neler yapılacağını belirleyebiliriz.

## BÖLÜM 5 – ARAÇLAR

## Delay (Erteleyici)



Sekil 121

Belirlenen bir süre sonunda bazı kod bloklarının çalıştırılması gerekebilir. Böyle bir durumda Delay (Erteleyici) nesnesini kullanırız. Örneğin, 5 saniye sonra ekranda video oynatılmasını istiyorsak delay nesnesi ile bunu kolaylıkla yapabiliriz. Delay nesnesinde en önemli unsurlardan biri Interval (Süre) özelliğidir. Özellikler

penceresinden Interval değerini ayarlayabiliriz. Bir diğer önemli özellik ise otomatik başlatma özelliğidir. Varsayılan olarak False (Yanlış) gelmektedir. Delay nesnesini, proje

başlarken çalıştırmak istiyorsak True (Doğru) yapmalıyız.

**NOT**: Delay nesnesinde varsayılan süre 1000 milisaniyedir bu da 1 saniyeye denk gelmektedir.

When Complete On Delay Event	On Delay Event (Erteleyici Olayı) bloğu ile delay nesnesi tamamlandığında, başladığında veya durduğunda neler yapılacağını belirleyebiliriz.
Delay Start Delay Control	Delay Control (Erteleyici Kontrolü) bloğu ile delay nesnesini başlatabilir veya durdurabiliriz.

## Timer (Zamanlayıcı)



Sekil 122

Projeye zamanlayıcı eklememizi sağlayan araçtır. Delay nesnesinde olduğu gibi timer'da da Interval (süre) ve otomatik başlatma özellikleri önemlidir. Interval zamanlayıcının hızını belirtir. Örneğin, interval değeri 3000 milisaniye olan bir zamanlayıcı 3 saniyede bir işler.

WhenTickOn Timer EventOn Timer Event (Zamanlayıcı Olayı) bloğu ile<br/>timer nesnesi başladığında, durduğunda veya<br/>süre her işlediğinde neler yapılacağını<br/>belirleyebiliriz.Timer StartTimer ControlTimer Control (Zamanlayıcı Kontrolü) bloğu<br/>ile timer nesnesini başlatabilir veya<br/>durdurabiliriz.



## Trigger (Tetikleyici)

Trigger nesnesi, bulunduğu anahtar karede çalışmaya başlar. Projede trigger bulunan anahtar kareye gidildiğinde, trigger bloğundaki kod blokları tetiklenir ve çalışırlar.

Şekil 123

ç

When Invokes On Invoke Event On Invoke Event Sahner neler y	voke Event (Tetiklenme Olayı) bloğu, ye yerleştirilen trigger tetiklendiğinde yapılacağı belirlememizi sağlar.
---	--





# **BÖLÜM 6 – SEBİT FRAMEWORK**

### **Sebit Framework**

Sebit Framework, özellikle eğitsel içerikler geliştiren kullanıcılar için oldukça faydalı blokları barındırır. Örneğin, soru doğru cevaplandığında doğru sesi dinletebilmek, soru gösterme veya aktivite yapıldığında küçük bir tamamlandı animasyonu göstermek bu bloklar aracılığıyla projeye eklenebilir. Böylelikle projemizdeki görsel ve işitsel etkileşimleri arttırabiliriz.

Sebit Framework'ü kullanabilmek için araç çubuğunda bulunan Sebit > Yardımcı adımlarını izlediğimizde karşımıza *Sebit Yardımcısı* isimli pencere çıkacaktır. Bu pencere aracılığı ile projemize bildirim, soru, tanıtım, animasyon penceresi ve hipotez ekleyebiliriz.

Sebit Yardımcısı		
Kategoriler Bildirimler Sorular Tanıtımı Geç Animasyon Penceresi Hipotezler	Yeni Bildirim Ekle	
	Kaydet İptal	

Şekil 124

Show Restart Button	Show Restart Button (Yeniden Başlat Düğmesini Göster) bloğu ile ekranımızın sol alt köşesine yeniden başlat butonu ekleyebiliriz.
Hide Restart Button	Hide Restart Button (Yeniden Başlat Düğmesini Gizle) bloğu ile ekranda eğer bir yeniden başlat düğmesi varsa onu gizleyebiliriz.
On Restart Event	On Restart Event (Yeniden Başlat Olayı) bloğu, sahnemizdeki yeniden başlat butonuna tıklandığında neler yapılacağını belirlememizi sağlar.



Show Logo Show Show Logo	Show Logo (Logoyu Göster) bloğu, sahnenin sol üst tarafına bir <i>Vitamin</i> logosu eklememizi sağlar.
Sco Complete	Sco Complete (Sco Tamamlandığında) bloğu, ekranda <i>tamamlandı</i> animasyonu oynatmamızı sağlar.
Play Correct1 Play Notify Sound Solo	Play Notify Sound Solo (Uyarı Sesini Solo Çal) bloğu yalnızca sesin çalınmasını sağlar.
When Notify Sound Hide Do	Notify Sound Hide Event (Uyarı Sesi Bitiş Olayı) bloğu ile uyarı sesi bittiğinde neler yapılacağını belirleyebiliriz.

Play Notify Sound (Uyarı Sesi Çal) bloğu, istenilen bir doğru veya yanlış sesi çaldıktan sonra soru veya geri bildirim gösterilmesini sağlar. Üc tane
---

### Feedbacks (Bildirimler)

*Sebit Yardımcısı* penceresinde Bildirimler başlığında *Yeni Bildirim Ekle* butonuna tıkladığımızda karşımıza *Bildirim Düzenle* penceresi gelecektir.

Bildirim Düzenle			
Id	1 🜩		
Bildirim Metini			*
Ses Url	1		x
Modal	Görünür	🗌 Otomatik Kapat	
			Kaydet İptal

Şekil 125


Bu pencereden eklenecek bildirimi ayarlayabiliriz. Burada Id numarasına dikkat edilmesi gerekiyor. Bloklar ile bildirime ulaşmak istediğimizde bu Id numarasını kullanırız. Bildirimi oluştururken Modal, Görünür, Otomatik Kapat özelliklerini ayarlayabilir ayrıca bildirime ses de ekleyebiliriz.

Show Feedback index Show Feedback	Show Feedback (Geri Bildirim Göster) bloğu, eklediğimiz bildirimin sahnede gösterilmesini sağlar. Index kısmında hangi bildirimin gösterileceği belirtilir.
Hide Feedback index	Hide Feedback (Geri Bildirim Gizle) bloğu, eklediğimiz bildirimin sahnede gösterilmemesini sağlar.
When Feedback Hide Do On Feedback Hide Event	On Feedback Hide Event (Geri Bildirim Gizlenme Olayı) bloğu ile bildirim sahnede gizlendiğinde neler yapılacağını belirtebiliriz.

#### **Questions (Sorular)**

*Sebit Yardımcısı* penceresinde Sorular başlığında *Yeni Soru Ekle* butonuna tıkladığımızda karşımıza *Soruları Düzenle* penceresi gelecektir.

Sorulari Duze	- me					
Id	1 ≑					
Soru Metni						-
						Ŧ
Ses Url						<b>x</b>
Seçenekler	r				<b>G</b>	△ ▽
Parça	Metin	Cevabı Varmı	Cevap Metni	Cevap Ses Dosyası		
				Kayde	t	İptal

Şekil 126

Bu pencere aracılığı ile soru ekleyebiliriz. Bloklar ile soruya ulaşmak istediğimizde Id numarası kullanırız. Seçenekler penceresinde istenilen kadar seçenek soruya ekleyebiliriz. Soruya ses ekleyebildiğimiz gibi doğru cevaba da ses ekleyebiliriz.



Show Question index Show Question	Show Question (Soruyu Göster) bloğu, oluşturduğumuz sorunun sahnede gösterilmesini sağlar. Index kısmında hangi sorunun gösterileceği belirtilir.
Hide Question index	Hide Question (Soruyu Gizle) bloğu, oluşturduğumuz sorunun sahnede gizlenmesini sağlar.
When Choice Selected Do On Question Choice Event	On Question Choice Event (Soru Seçenek Olayı) bloğu ile seçenekler seçildiğinde neler yapılacağını belirleyebiliriz.
When Hide Do On Question Hide Event	On Question Hide Event (Soru Gizleme Olayı) bloğu ile soru gizlendiğinde neler yapılacağını belirtebiliriz.

#### Skip Intro (Tanıtımı Geç)

*Sebit Yardımcısı* penceresinde Tanıtımı Geç başlığına tıkladığımızda projemize kolaylıkla bir tanıtım videosu ekleyebiliriz. Eklediğimiz tanıtım projemizden önce sahneye gelir.

Sebit Yardımcısı				
Kategoriler Bildirimler Sorular Tantımı Geç Animasyon Penceresi Hipotezler	Skip Intro Kulla Skip Intro Tipi Video Dosya Adı	n Animation Animation Intro	· · · · · · · · · · · · · · · · · · ·	
				Kaydet İptal



#### Animation Popup (Animasyon Penceresi)



Animasyon Penceresi projemize video ve animasyonlar eklememizi sağlar. Dizayn bölümünde Sebit Framework başlığından sürükle bırak yöntemiyle sahnemize ekleyebiliriz. Animasyonu, *Sebit Yardımcısı* aracılığı ile ekleriz fakat öncesinde animasyon penceresinin sahneye eklenmiş olması gerekmektedir.

Şekil 128

Sebit Yardımcısı penceresinde Animasyon Penceresi başlığında Yeni Animasyon Penceresi Ekle butonuna tıkladığımızda karşımıza Animation Popup Düzenle

penceresi gelecektir.

Animation Popu	p Düzenle			
Id	1 ≑			
Video Url				<b>x</b>
Modal	C Kontroller	Otomatik Kapat	🗌 Otomatik Oynat 📄 Kapat Buto	nu 🗌 Çerçeve
Genişlik	1			
Yükseklik	1			
			Kayd	et Iptal

Şekil 129

Açılan pencere aracılığıyla projemize video ekleyebiliriz. Eklediğimiz videoya Id numarası ile bloklardan ulaşabiliriz. Videoya ayrıca Modal, Kontroller, Otomatik Kapat/Oynat, Kapat Butonu, Çerçeve özelliklerini ekleyebilir ve genişlik ile yüksekliğini de belirleyebiliriz.

Show Animation Popup in Show Animation Popup in	how Animation Popup how Animation Popup Show Animation Popup (Pencere Animasyonu Göster) bloğu, eklediğimiz animasyonun sahnede gösterilmesini sağlar. Index kısmında hangi animasyonun gösterileceği belirtilir.
Hide Animation Popup in	Hide Animation Popup (Pencere Animasyonunu Gizle) bloğu, eklediğimiz animasyonun sahnede gizlenmesini sağlar.



When Con Animation Popup Event	On Animation Popup Event (Pencere Animasyon Olayı) bloğu ile eklediğimiz animasyon sonlandığında (Video Ended) veya kapatıldığında (Video Close) neler yapılacağını
	belirleyebiliriz.

#### Hypothesis (Hipotezler)

*Sebit Yardımcısı* penceresinde Hipotezler başlığında *Yeni Hipotez Ekle* butonuna tıkladığımızda karşımıza *Hipotezi Düzenle* penceresi gelecektir.

Hipotez Düzenle	e		
Id	1 🜩		
Soru Metni			A
			Ψ
Tahmin Metni			A
	Karaka Katalari		·
	Kombo Kutulari	Kombo Kutusu Ob	vieleri
			Yeni Obie Ekle
			Obje Sil
			Obje Değiştir
			Obje İsmi
			Obje Değeri
	Genişlik: 700	\$	Üst Pozisyon: -1 🛊
	Yükseklik: 340	\$	Sol Pozisyon: -1 🗘
			Varsayılan yerde çıkması için,
	Tahmin metni alanına con yere {{combo1}} yazara	ibobox kullanabilmek iç k combo1 adında bir co	in combobox kullanmak istediğiniz pmbobox yaratabilirsiniz.
			Kaydet İptal

Şekil 130

Hipotez Düzenle penceresi ile projemize hipotez ekleyebiliriz. Eklediğimiz hipoteze Id numarası ile bloklardan ulaşabiliriz.



Show Hypothesis index Show Hypothesis	Show Hypothesis (Hipotezi Göster) bloğu, eklediğimiz hipotezin sahnede gösterilmesini sağlar. Index kısmında hipotezin Id numarasını belirtilir.
Hide Hypothesis index	Hide Hypothesis (Hipotezi Gizle) bloğu, eklediğimiz hipotezin sahnede gizlenmesini sağlar.
On Hypothesis Click Event Do On Hypothesis Click Event	On Hypothesis Click Event (Hipotez Tıklama Olayı) bloğu ile hipoteze tıklanıldığında neler yapılacağını belirleyebiliriz.
On Hypothesis Changed Event Do On Changed Event	On Hypothesis Changed Event (Hipotez Değişme Olayı) bloğu ile hipotez değiştiğinde neler yapılacağını belirleyebiliriz.



## BÖLÜM 7 – HAZIR KÜTÜPHANE FONKSİYONLARI

### Math (Matematik)

Matematik kütüphanesi, matematiksel işlemlerde kullandığımız blokları barındırır. Aritmetik işlemlerin yanı sıra, karşılaştırma işlemleri de bu kütüphanede bulunur.

Value	Value (Değer) bloğu, tam sayı değeri eklememizi sağlar.
Float Value	Float Value (Ondalıklı Değer) bloğu, ondalıklı bir sayı değeri eklememizi sağlar.
🔅 < 🕂 + < 🗾 Sum	Sum (Topla) bloğu, iki veya daha fazla sayının toplamını döndürür.
Subtract	Substract (Çıkar) bloğu, iki sayının farkını döndürür.
Aultiply	Multiply (Çarp) bloğu, iki veya daha fazla sayının çarpımını döndürür.
Divide	Divide (Bölme) bloğu, iki sayının bölümünü döndürür.
Power	Power (Üssü) bloğu, ilk eklenen sayının ikinci eklenen sayı kadar üssünü döndürür.
Equality	Equality (Eşitlik); iki sayı arasındaki eşit, eşit değil, büyük eşit, küçük eşit, büyüktür ve küçüktür ifadelerinin sonucunu döndürür.
fixed decimal places by 2 Fixed Decimal Places	Fixed Decimal Places (Sabit Ondalık Değer) bloğu, ondalık değere sahip bir sayının virgülden sonra kaç hane olduğunu döndürür.
random float from O to O decimal point digits Random Float	Random Float (Rastgele Ondalık Değer) bloğu, eklenen iki değer arasından rastgele bir ondalık değer döndürür.



S	а	y	f	а	I	78
---	---	---	---	---	---	----

random integer from 0 to 100 Random Integer	Random Integer(Rastgele Tamsayı) bloğu eklenen iki değer arasından rastgele bir tam sayı değeri döndürür.
	To Numeric (Sayıya Çevir), eklenen bloğu sayıya çevirir.
round	Round (Yuvarla) bloğu, eklenen değeri matematiksel değerine göre yuvarlar.
Floor	Floor (Taban) bloğu, eklenen sayıdan küçük veya ona eşit olan en büyük tam sayıyı döndürür.
Min Max	Min Max (Minimum Maksimum) bloğu, eklenen iki değer arasındaki en büyük veya en küçük değeri döndürür.
Evaluate	Evaluate (Değerle) bloğu, JavaScript'teki eval methodu olarak da bilinir. Bu blok aracılığıyla, eklediğimiz karakter dizisini JavaScript kaynak kodlarına çevirir.
Map Value fromLow toLow fromHigh 100 toHigh 255 Map	Map (Oranla) bloğu eklediğimiz değerler arasında oranlama yapmamızı sağlar. Örneğin 0 ile 10 arasında bulunan 5 sayısının 0 ile 100 arasına oranladığımızda hangi sayı olduğunu öğrenmek istediğimiz bu bloğu kullanabiliriz.

### Math Advanced (İleri Matematik)

Daha ileri matematiksel işlemler için İleri Matematik blokları kullanılabilir. Örneğin; logaritma, mutlak değer, sinüs, kosinüs gibi bloklar burada yer alır.

Absolute	Absolute (Mutlak Değer) bloğu, eklenen sayının mutlak değerini döndürür.
Atan	Atan bloğu, eklenen sayının arctanjantını derece cinsinden döndürür.



Atan2 , Atan2	Atan2 bloğu, eklenen iki sayının arctanjantını döndürür.
Asin	Asin bloğu eklenen sayının arcsinüsünü derece cinsinden döndürür.
Ceil	Ceil (Tavan) bloğu, eklenen sayıdan büyük veya ona eşit olan en küçük tam sayıyı döndürür.
Cos	Cos bloğu, eklenen sayının kosinüsünü derece cinsinden döndürür.
Ехр	Exp bloğu, eklenen sayıyı eksponansiyel sayısının (2,71828) üssü olarak döndürür.
	Log (Logaritma), eklenen sayının
	logaritmasını döndürür.
Sin Sin	Sin (Sinüs), eklenen sayının sinüs değerini derece cinsinden döndürür.
Sqrt Sqrt	Sqrt (Karekök), eklenen sayının karekök değerini döndürür.
Tan	Tan bloğu, eklenen sayının tanjant değerini derece cinsinden döndürür.
Constant Constant	Constant (Sabit), bazı önemli sayıların (örn: e, pi) sabit değerlerini döndürür.
Minumum In List	Minimum in List bloğu, listedeki en küçük değeri döndürür.
Maximum In List	Maximum in List bloğu, listedeki en büyük değeri döndürür.
Mod f: Modulus	Modulus (Mod) bloğu, eklenen sayının modunu döndürür. Hangi sayıya göre modunun alınacağını da belirleyebiliriz.
	ls NaN (Sayı Değil mi) bloğu, eğer eklenen nesne bir sayı ise True (Doğru) değerini döndürür aksi hâlde (Yanlış) değerini döndürür.
Plus One Plus One	Plus One (Bir Artır) bloğu, eklenen değeri bir arttırır.



	Minus One (Bir Azalt) bloğu, eklenen
Minus One Minus One	değeri bir azaltır.

## **Trigonometry (Trigonometri)**

İleri Matematiğe ek olarak, VFabrika'da trigonometrik işlemleri yapmamızı sağlayan bloklar da yer alır.

Degree To Radian Degree To Radian	Degree To Radian (Dereceyi Radyana Çevir) bloğu, eklenen değeri radyana çevirir.
Radian To Degree Radian To Degree	Radian To Degree (Radyanı Dereceye Çevir) bloğu, eklenen değeri dereceye çevirir.
Create Vector 2D X O , Y O Vector 2D Create	Vector 2D Create (2B Vektör Oluştur) bloğu, girilen X ve Y değerlerine göre bir vektör oluşturur.
Vector 2D Add + Vector 2D Add	Vector 2D Add (2B Vektör Topla) bloğu, iki vektörün toplamını döndürür.
Vector 2D Subtract	Vector 2D Substract (2B Vektör Çıkart) bloğu, iki vektörün farkını döndürür.
Vector 2D Multiply	Vector 2D Multiply (2B Vektör Çarp) bloğu, vektörün eklenen sayı ile olan çarpımını döndürür.
Vector 2D Divide	Vector 2D Divide (2B Vektör Böl) bloğu, vektörün eklenen sayı ile bölümünü döndürür.
Vector 2D Inverse	Vector 2D Inverse (Tersine Çevir) bloğu, vektörü tam tersi yönüne çevirir. Örneğin, Vektör X: 10, Y: 20 ise tersine çevir bloğu ile X: - 10, Y: -20 olur.



Vector 2D Set Length	Vector 2D Set Length (Uzunluğu Değiştir) bloğu, vektörün eklenen sayı kadar uzunluğunu değiştirir. Vektör aynı yönde fakat farklı büyüklükte olur.
Vector 2D is Zero Vector	Vector 2D is Zero Vector (Boş 2B Vektör mü?) bloğu, eklenen vektörün sıfır olup olmadığını kontrol eder. Vektör eğer sıfır ise True (Doğru), değilse False (Yanlış) değerini döndürür.
Vector 2D Get Length	Vector 2D Get Length (Uzunluk Hesapla) bloğu, vektörün uzunluğunu döndürür.
Vector 2D Normalize	Vector 2D Normalize (Birim Vektöre Çevir) bloğu, vektörü birim vektörüne dönüştürür. Vektörün uzunluğu 1 olur ve yönü değişmez.
Vector 2D Dot Product	Vector 2D Dot Product (Nokta Çarpımı) bloğu, eklenen iki vektörün nokta çarpımını döndürür. Nokta çarpımı, karşılıklı bileşenlerin çarpımları toplamına eşittir.
Vector 2D Cross Product	Vector 2D Cross Product (Çapraz Çarpım) bloğu, eklenen iki vektörün çapraz çarpımını döndürür.
Vector 2D Distance	Vector 2D Distance (Uzaklık) bloğu, eklenen iki vektör arasındaki uzaklığı döndürür.
Vector 2D Rotate by 90 degree Vector 2D Rotate Degree	Vector 2D Rotate Degree (Açıyla Döndür) bloğu, eklenen vektörü belirtilen açı doğrultusunda döndürür.



Vector 2D Angle 180 With Signed Vector 2D Angle	Vector 2D Angle (2B Vektör) bloğu, 180 derece işaretli veya işaretsiz iki vektör arasındaki açıyı döndürür.
Vector 2D Equality == Vector 2D Equality	Vector 2D Equality (2B Vektörel Eşitlik) bloğu, eklenen iki vektörün eşit olup olmadığını kontrol eder. Vektörler birbirine eşitse True (Doğru), değilse False (Yanlış) değerini döndürür.
Vector 2D Center Of Points	Vector 2D Center of Points (Noktanın Ağırlık Merkezini Hesapla) bloğu, vektörlerin ağırlık merkezini döndürür. Örneğin, bir listede 4 tane vektör varsa bu vektörlerin ağırlık merkezini verir.
Vector 2D Sort Points by Angle	Vector 2D Sort Points by Angle (Noktaları Açıyla Göre Sırala) bloğu, eklenen listede yer alan vektörleri X ekseninden başlayarak sıralar.
Vector 2D Area Of Points	Vector 2D Area of Points (Noktaların Oluşturduğu Alanı Hesapla) bloğu, eklenen listede yer alan vektörlerin toplam alanını döndürür.

#### **Json Data Set**

Json bir veri değişim formatıdır. Daha küçük boyutlarda veri saklamak ve alıp göndermek için kullanılır. VFabrika'da bir JSON DataSet Düzenleyici bulunmaktadır. Araçlar > Json veri seti editörü yolu ile ulaşılabilir.



Buradan bir tablo oluşturabilir ve onu da bir veri tabanı gibi kullanabiliriz. Bloklar yardımıyla oluşturduğumuz tablodan veri çekebiliriz. Ancak, veri ekleme veya değiştirme işlemlerini yapamayız.

Get Value from <b>tableName</b> at row <b>O</b> . <b>FieldName</b> Get Value		
Get Value (Değer Oku) bloğu, oluşturduğumuz tablonun adını, satırını, alan adını girerek istediğimiz veriye ulaşmamızı sağlar.		
Get row from <b>TableName</b> at index <b>Get Table Row As Object</b>		
Get Table Row As Object (Tablo Satırını Nesne Olarak Al) bloğu, tablo adını ve satırını girdiğimiz verileri bir nesne olarak bize getirir.		
Get Row Count of tableName Get Table Row Count		
Get Table Row Count (Tablo Satır Sayısı) bloğu, oluşturduğumuz tablodaki satır sayısına ulaşmamızı sağlar.		



# BÖLÜM 8 – DİĞER ARAÇLAR

#### General

Create Object	Create Object (Nesne Oluştur) bloğu VFabrika'da nesne oluşturmamızı sağlar. Nesneyi oluşturmak için değişken tanımladığımızda değer kısmına Create Object bloğunu ekleriz.
Set Z Index	Set Z Index (Z Index Ayarla) bloğu seçtiğimiz nesnenin z-index'ini yani sahnedeki hiyararşik durumda önde veya geride olmasını sağlar.
Get Z Index	Get Z Index (Z Index Oku) bloğu seçtiğimiz nesnenin z-index sayısını öğrenmemizi sağlar.
On App Initializing On App Initializing	On App Initializing (Uygulama Başlıyor Olayı) bloğu uygulamamız başlarken çalışmasını istediğimiz blokları çalıştırır.
On App Initialized On App Initialized	On App Initialized (Uygulama Başladı Olayı) bloğu uygulamamız başladıktan sonra çalışmasını istediğimiz blokları çalıştırır. Burada dikkat etmemiz gereken konu, On App Initializing ve On App Initialized blokları Timeline Event bloğundaki Initialize seçeneğinden her zaman daha önce çalışırlar.
On App Deinitialize On App Deinitialize	On App Deinitialize (Uygulama Bitti Olayı) bloğu uygulamamızı sonlandırdığımızda çalışmasını istediğimiz blokları çalıştırır.







Map Folder Tolder Mapper	Folder Mapper (Klasör Eşleme) bloğu projemize dinamik olarak bir dosya ekleme yapmak istediğimizde
Set Property	Set Property (Özellik Yaz) bloğunda, ilk açılır pencerede sahnedeki tüm nesneleri görebilir ve ikinci açılır pencerede de seçtiğimiz nesnenin tüm özellikleri listelenir. Burada istediğimiz nesnenin özelliğini ayarlayabiliriz.
Get Property Get Property	Get Property (Özellik Oku) bloğu ile tüm eklediğimiz nesnelerin özelliklerine ulaşabiliriz.
Get Reference	Get Reference (Referans Oku) bloğu, sahnedeki nesnelere ulaşmamızı sağlar. Get Property (Özellik Oku) bloğunda da Get Reference bloğu ile nesneye ulaşabiliyoruz.
Get Reference Property	Get Reference Property (Referans Özelliği Oku) bloğu, nesnelerin özelliklerine ulaşmamızı sağlar. Get Property (Özellik Oku) bloğunda Get Reference Property bloğu ile nesnenin özelliğine ulaşabiliriz.
Get Attribute	Get Attribute (Nitelik Oku) bloğu ile niteliğini okumak istediğimiz nesneyi seçtikten sonra yazdığımız niteliğe ulaşabiliriz.
Set Attribute	Set Attribute (Nitelik Yaz) bloğu ile niteliğini ayarlayacağımız nesneyi seçtikten sonra nesnenin eski ve yeni niteliğini yazarak nesnenin niteliğini değiştirebiliriz.



Туре	Type (Tür) ile VFabrika'da bulunan bütün nesne türlerini seçebiliriz.
Cursor Default Cursor	Cursor (İmleç) bloğu ile nesnelere istediğimiz imleç türünü ekleyebileceğimiz bloktur. Ekleyebileceğimiz türler Not Set (Değiştirme), Default (Varsayılan), None (Yok), Pointer (El), Crosshair (Artı), Move (Hareket), Text (Metin).

#### Animation



Animate Design Object (Dizayn Nesnesine Animasyon Yap) bloğu ile istediğimiz nesneye animasyon ekleyip o animasyonu da başlatabiliriz. Açılır kapanır menüden animasyon eklemek istediğimiz nesneyi seçeriz. Animasyonun özelliklerini belirlemek için sol taraftaki çarka tıklayarak gelen seçeneklerden yapmak istediğimiz animasyona göre seçim yapabiliriz. Örneğin X ve Y'yi seçerek nesnenin pozisyonunu animatif olarak değiştirebiliriz. Ardından animasyonumuza isim verebiliriz. Burada dikkat etmemiz gereken konu, Animasyon başlığı altındaki bloklarla eklediğimiz animasyonlar, Zaman Çizelgesi animasyonları ile karıştırılmamalıdır.

Easing Function Linear Easing Function	Easing Function (Hareket Tipi) bloğu ile animasyonumuzun nasıl bir hareket şekline olacağını belirleyebiliriz.
Stop Animation Stop Animation	Stop Animation (Animasyonu Durdur) bloğu ile ismini yazdığımızı animasyonu durdurabiliriz.
Stop All Animations	Stop All Animations (Tüm Animasyonları Durdur) ile uygulamamızdaki tüm animasyonları durdurabiliriz.



When VV Do Design Object Animation Event	Design Object Animation Event (Dizayn Nesnesi Animasyon Olayı) bloğu seçtiğimiz nesnedeki bir animasyonun başladığında, devam ettiğinde, durduğunda veya bittiğinde çalışmasını istediğimiz blokları çalıştırır.
General Animation Event	General Animation Event (Genel Animasyon Olayı) bloğu uygulamamızdaki herhangi bir animasyon başladığında, devam ettiğinde, durduğunda veya bittiğinde çalışmasını istediğimiz blokları çalıştırır.
Default Fps	Set Default Fps (Varsayılan Fps'i Ayarla) bloğu animasyondaki varsayılan Frame Per Second (Saniyedeki Kare Hızı) ayarlamasını yapabildiğimiz bloktur.
Default Duration	Set Default Duration (Varsayılan Süreyi Ayarla) bloğu ile animasyonlar için varsayılan süre ayarlaması yapabiliriz.
Default Easing Function Linear Set Default Easing Function	Set Default Easing Function (Varsayılan Hareket Tipini Ayarla) bloğu ile animasyonlardaki varsayılan hareket tipini ayarlarız.



## **BÖLÜM 9 – ŞABLONLAR**

VFabrika programı açıldığında "Yeni Proje Oluştur" sayfasında bulunan "Şablonlar" seçeneği seçilir. Sol tarafda çıkan menüden istenen etkinlik şablonunu seçerek ilerleriz. Açılan şablonlarda F12 ile projeyi yayınlayıp, demo proje incelenebilir. Demo proje üzerinden değişiklik yaparak kendi etkinliğinizi hazırlayabilir ve tekrar F12 yaparak çalıştırabilirsiniz.

Y. Isimstr - Wabrika - Sebit - Tüm hakdarı saklıdır. Doşya Düzenle Görünüm Proje Yayınla Azeçlar Yardım	X. \$
Destance Bix bald bio 6 90	
Baglangç Sayfaa 🛪	
VFabrika Gelgtme Ottam Başlangıç Yen Pide Oluştur Pide Ac	
Önceki Öğeler	Yeni Proje
Projentyk Cri Waren (do Wesktop Projentyk, Projentyk, vef	Veni Proje Obsturr Bu drien in jernit bri prije obstanditrinz.
Projentiklyk C: (Lawes) (dd Denktop) Projentiklyk (Projentiklyk, vaf	
Projemparalaida	Sobolar Carbona Strong (Diu) A Addama Karbona Shore (In protection Shore
denens03_12020	Diger Bogladem Daldar (Agir Kunder) /
C: (pres/t/dc/pres/top/deneme08_12020/deneme08_120 deneme	Sogakian Dokker Naukan)
C:/Users/tids/Desktop/deneme/deneme.vsf	200em Kartelar Proges (Oku)
Ci (Deers (tids / Desktop / Karanik Oda/r duno / Karanik Oda/r	Edmeys Tahmin El Projest (Dau)
ek_pensou C:\Users\\ids\Desktap\tak_pensou\sk_pensou.vsf	Logikkink Epiegtime Projes (3
nemSensoruArdunio Crijušem (tda) Posktopi (nemSensoruArdunio (nemSensoruAr	Conseler Ubernde Eglestme Pr
	Noje de Projem
	Konun C-Laesiblo Pestop
	Cotin Ad Projem
	Greek Glegar lets
Data	

Şekil 132

#### **Kare Bulmaca**

- Bu şablonda verilen cümlelere ait bilinmeyen kelimeleri kullanıcı bulmacada seçerek bulması istenmektedir.
- Sol tarafta verilen cümlelerdeki kelimelere tıklayarak varsa kelimeye ait ipucu gösterme özelliği bulunur. İstenirse bu özellik kapatılabilir.
- Kelimeler bulunduğunda her kelime farklı bir renge karşılık gelecek şekilde gösterilir ve doğru/yanlış cevaba dair ses effekti verilmiştir.



nd the words which is hidden horizontally and vertically. 00:00							<b>©</b> •)							
F000-11/0 10/-				68	1.2		Ľ	<u></u>	1	Y	1	2	2	1
SELFISH	K	Ö	Ν	ş	U		В	Ν	F	R	U	Ğ	С	13
	J	U		V	D	J	i	J	J	E	Ğ	E	R	DC
WINDING SPOOL IN A FISHING ROD	Ö	Α	Y	J	Ρ	Ş	V	A	М	E	R	Ö	C	F
BEFORE A GIVEN TIME	Ö	Ü	Y	U	N	Т	I	L	U	L	Μ	R	Н	51)
	Y	С	F	G	Т	N	L	V	С	F	L.	С	E	主
> DAIRY PRODUCTS, CAN BE HARD OR SOFT	E	н	0	Ν	L	Ι	Ν	E	С	J	Ρ	Ğ	E	J
	С	Ş	i	Ρ	J	Ι	S	G	R	D	A	G	S	TTT
Turunt	U	R	Α	Т	С	i	С	в	Ν	Ρ	R	0	E	75
PUBLIC OPEN SPACE	E	0	н	z	Ü	J	i	R	D	Ç	ĸ	N	S	2
	Ş	U	i	Ü	V	Ş	L	Α	Y	L	Ç	F	Ğ	H
	I	N	С	0	Ν	S	I.	D	E	R	A	Т	E	9
	V	Μ	Ρ	Т	0	κ	H	С	J	Ç	Y	Ç	Ö	
	1	L	Y	κ	Ü	Y	J	М	Ç	i	P	A	S	1
			5		1			1.	. (	1	N		Fo	202
E/ E) a / A/ A						L	-	5	16	[d	10	71	12	0000
						Y	D	, C	7	à. '	-	10	L	
					1	λĝ		-		1.	5	ħ		

Sayfa **| 90** 

Şekil 133

	N	Y	G	н	0	N	ľ		N	Y	N	G	C
SELFISH	C	c	i	Z	G	M	С	D	U	Ē	Ü	E	G
WINDING SPOOL IN A FISHING ROD	v V	Z	K	N	G	С	н	Z	J	С	Ğ	ī	С
BEFORE A GIVEN TIME	R	R	i	L	Т	Y	E	F	Y	J	Ρ	С	A
	E	S	Ş	F	S	۷	E	F	Y	Ρ	A	Ζ	В
DAIRY PRODUCTS, CAN BE HARD OR SOFT	E	Ğ	L	V	S	R	S	Ö	S	S	R	F	L
> OVER THE WEB	L	В	Т	Ν	A	S	E	В	0	В	К	Ν	U
	i	Ö	С	D	D	Т	S	М	Ζ	Ν	С	i	۷
> PUBLIC OPEN SPACE		N	С	0	Ν	S	1	D	Е	R	A	Т	E
	D	K	R	J	Κ	Κ	U	В	L	V	D	0	Ğ
<u>3</u>	С	S	Т	М	Ş	E	Ν	0	Ν	В	S	Ğ	Z
	1	Ç	В	Ρ	U	Α	Y	D	S	F	L	D	Ü
01/101	V	E	Ü	U	Ν	Т	I.	L	A	в	Ğ	Α	F
								11.	6	31	N		F



#### **Boşlukları Doldurma (Açılır Kutular)**

- Bu şablonda paragrafta bulunan boşlukların verilen sese göre doğru cevabının seçilmesi beklenmektedir. Kullanıcı boşluklara ait seçenekleri açılır-kapanır menüden seçer.
- Doğru ve yanlış cevaplar renklendirilir.





#### Sayfa **| 92**





#### Boşlukları Doldurma (Metin Kutular)

- Bu şablonda paragrafta bulunan boşlukların verilen sese göre doğru cevabının metin kutularına girilmesi beklenmektedir.
- Doğru ve yanlış cevaplar renklendirilir.

Match the verbs with phrases about free time activities.	00:00 🔹
Hello, everybodyl My name is Julia. I am because it is my birthday today. We are in the garden for is my best friend. She is and today. She is helping my mother with the compared by the shelping my my my my my my my my my my my my my	my party. Linda decorations. Jack
is my classmate. He is walking around and I think he feels . Betty is my classmate, too. She is talking and she looks . Oh, Saml My lovely brother! He is . and he	is also my guest
Check your answers	SON BOR



#### Sayfa **| 94**

Match the verbs with phrases about free time activities.	19:02 💿
Hello, everybodyl My name is Julia. I am excited because it is my birthday today. We are in the garden for my party. is my best friend. She is happy and energetic today. She is helping my mother with the decoration is my classmate. He is walking around and I think he feels bored. Betty is my classmate, too. She is talking with my and he looks and he looks	Linda s. Jack y sister
drink some water. Sally is my cousin. She is hungry and she needs to eat today. He is tired and sleepy . He needs to have a rest.	y guest
Check your answers	

Şekil 139





#### Dönen Kartlar Projesi

- Bu şablon eşleştirme yapılacak ürünler için kullanılabilir.
- Bu şablon ile metin-metin, resim-metin ve resim-resim eşleştirmesi yapılabilir.



#### Şekil 141





#### Kelimeyi Tahmin Et Projesi

- Bu şablonda verilen açıklamaya göre istenilen kelimenin harfleri bulunur. Kullanıcı İngilizce veya Türkçe harf grubundan birini seçebilir. Seçilen harflere tıklayarak bulma işlemi yapılır.
- Verilen yıldızların tamamı bitince doğru cevap gösterilir ve bir sonraki soruya geçilebilir.
- Son ekranında verilen sorulara ait cevaplar ve kullanıcının aldığı yıldız sayısı gösterilir.







Şekil 144

#### Boşluklarla Eşleştirme Projesi

- Bu şablonda verilen resimlere göre istenilen kelimeler metin kutularına yazılır,
- Doğru ve yanlış cevaplar renklendirilir.





#### Sayfa **| 98**



Şekil 146







## Görseller Üzerinde Eşleştirme Projesi

- Bu şablonda verilen kelimeler uygun resimler ile eşleştirilir.
- Doğru ve yanlış cevaplar renklendirilir.









Şekil 149





### Çoktan Seçmeli Projesi

• Verilen paragrafa göre doğru cevaplar seçilir.



Şekil 151

#### Cümle Sıralama Projesi

- Bu şablonda verilen sese göre cümleler sıralanır.
- Doğru ve yanlış cevaplar renklendirilir,
- 3 yanlıştan sonra doğru cevaplar gösterilir.



Listen and put	: the phrases into the correct order.	00:08	
	How are you today?		
	Are you ready for the lesson? Fine, thanks and you?		
	Good morning, Mrs. Brown.		
1 Lail Di	Good morning, everybody. Be quite please.		1505
	Thank you.		H.
	Yes!		置い
2001/18	Sit down, please.		
	LA CONTRACT	50	

<u>Şekil 152</u>

### Dönen Tekerlek Projesi

- Bu şablonda açık uçlu sorulabilecek sorular girilir,
- Öğretmenin sınıf içinde kullanması için tasarlanan bir şablondur.
- Herhangi bir cevap girme alanı bulunmaz.





