

Table of Contents



Intermediate 1		
Meeting 1	How Cool is Programming World?.....	11
Meeting 2	Number Generator.....	21
Meeting 3	Event on Quiz.....	38
Meeting 4	Guessing Game.....	50
Meeting 5	Conditional Loop in Code Combat.....	63
Meeting 6	Internet Return Value.....	71
Meeting 7	JavaScript Heap and Stack.....	87
Meeting 8	Middle Test.....	99
Meeting 9	Artificial Intelligence with Match Game.....	100
Meeting 10	Media, Your Brain and Expression.....	119
Meeting 11	Digital Tracking and Array.....	134
Meeting 12	This is Object.....	157
Meeting 13	Currated, Object Method, and Create Canvas.....	164
Meeting 14	JavaScript Classes and Snake Game.....	191
Meeting 16	Online Hate Speech & Deploy	206
Meeting 16	Intermediate 1 Test.....	223
Intermediate 2		
Meeting 1	Phaser As A Framework.....	224,
Meeting 2	Collecting Stars: Making Game Environment.....	242
Meeting 3	Collecting Stars: Adding Physics and Score.....	259
Meeting 4	Bunny Jump: Follow the Player.....	237
Meeting 5	Bunny Jump: Scrolling Platform.....	288
Meeting 6	Bunny Jump: Collect the Carrot.....	303
Meeting 7	Bunny Jump: Handling Game Over.....	317
Meeting 8	Corona Buster: Moving Clouds and Add Button.....	327

1

HOW COOL IS PROGRAMMING WORLD?



Apa Yang Akan Kita Pelajari?

1. Pengenalan elemen dalam halaman web
2. Membuat halaman web pertama



HOW COOL IS PROGRAMMING WORLD?



Check This Video



Hello, World! Award Winning Short Film (Shot on GH4)

 Apa yang kalian pikirkan setelah menonton video ini? Menurut kalian mungkin tidak remaja seperti kalian sukses menjadi **App/ Game Developer?**



Berkenalan dengan Apps/Game Developer

1

Nick D'Aloisio

16 years old.

Founder of :



Summly



Summly adalah aplikasi untuk meringkas artikel/ berita bahasa inggris. Tahun 2013 diakuisisi Yahoo seharga 30 M USD (4 Triliun Rupiah)

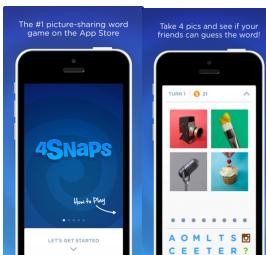


2

Michael Sayman

Start from 12 years old.

Founder of : **4Snaps**



Aplikasi game tebak kata dengan 4 gambar. Di umur 17 tahun sudah magang di Facebook diminta Mark Zuckerberg.

3

Arjun Kumar

14 years old

Winner MIT App Inventor Contest

Founder of : **EZ School Bus Locator**



Aplikasi untuk tracking School Bus agar orang tahu keberadaan anak.



•WHAT IS JAVASCRIPT?



What is JavaScript?



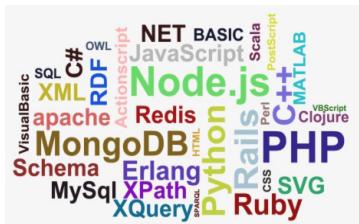
Tahukah kamu, kalau ada banyak bahasa pemrograman text?



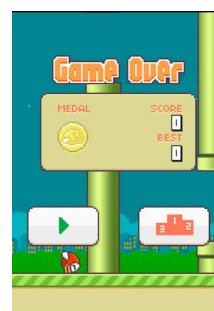
Vocabulary

JavaScript

Salah satu bahasa pemrograman yang populer dipakai untuk membuat websites lebih interaktif dan bisa juga untuk games.



Apa saja yang bisa dibuat dengan JavaScript?





DID YOU KNOW?

JavaScript dikembangkan pada 1994 yang dibuat dan didesain oleh **Brendan Eich**. Bahasa ini awalnya bernama Mocha, lalu diganti menjadi Mona, kemudian menjadi Livescript lalu resmi menjadi Javascript.

Tahun 2016, 92% website dikembangkan menggunakan JavaScript, menjadikannya salah satu bahasa penting bagi web developer.



Program in Real Web



Tahukah kamu, sebenarnya bagaimana cara kerja JavaScript?

subscribe



JavaScript membuat web page HTML bisa menjadi dinamis dan interaktif. **Misalnya**, ketika menekan tombol subscribe di YouTube maka tombol akan berubah menjadi subscribe dan jumlah subscribenya jadi bertambah!

Tetapi tombol subscribe tidak dibuat dengan JavaScript, tombolnya dibuat menggunakan HTML.



Semua halaman web yang sering kalian buka, seperti facebook, twitter, google, bahkan website Timedoctor ditampilkan dengan **menggunakan HTML**.

Game Javascript yg akan dibuat juga perlu **HTML** loh agar bisa ditampilkan di website !



Intro to HTML

Kali ini kita akan membuat simple number generator dalam sebuah web page. Untuk dapat menampilkan dalam web , kita perlu berkenalan dengan HTML



Vocabulary

HTML (Hyper Text Markup Language)

Bahasa markup (penanda) berbasis text , juga disebut sebagai formatting language (bahasa untuk memformat) yang digunakan untuk membuat sebuah halaman web.

HTML terdiri dari berbagai elemen, elemen akan memberikan perintah untuk menampilkan berbagai informasi di sebuah web.



Vocabulary

Elemen HTML

Elemen HTML didefinisikan dengan tag awal, isi elemen, dan tag akhir. `<tagname>Content goes here...</tagname>`

HTML juga memiliki aturan dalam penulisannya, ada beberapa elemen yang harus dituliskan dan sudah menjadi ketentuan. Ini adalah aturan dasar dalam penulisan dokumen HTML.

Syntax!

```
<!DOCTYPE html>
<html>
    <head>
        <title>
            // web page title here
        </title>
    </head>
    <body>
        // web page body here
    </body>
</html>
```



Pertemuan 1-How Cool is Programming World? HTML SCRIPT TAG

HTML <script> Tag

```
<html>
  <body>
    <script>
      console.log('Hello world!');
    </script>
  </body>
</html>
```

Untuk menambahkan JavaScript ke dalam file HTML digunakan tag **<script> . . . </script>**.

Syntax!

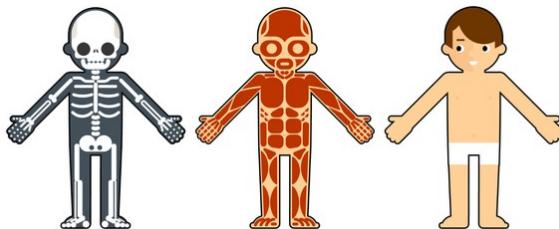
```
<!DOCTYPE html>
<html>
  <head>
    <title>
      // web page title here
    </title>
  </head>
  <body>
    // web page body here
  </body>
  <script>
    // javascript code here
  </script>
</html>
```



Selain HTML dan JavaScript, diperlukan juga CSS dalam web development. CSS adalah bahasa Cascading Style Sheet, yang biasanya digunakan untuk mengatur style dari HTML. Tapi kita akan lebih banyak belajar di JavaScriptnya saja!



HTML, JS, CSS



HTML

The Skeleton

JS

The Brain &
Muscular System

CSS

The Skin



Pernahkah kalian melihat programmer mengetik program?
Menurutmu, dimana programmer mengetik program tersebut?

Saat di Beginner, program dibuat di Scratch /CoSpaces/ Tynker atau Playcode. Nah, sekarang saatnya mulai menggunakan teks Editor!



Vocabulary

Teks Editor

Software yang digunakan oleh programmer untuk menulis program. Contohnya Visual Studio Code, Atom, Sublime Text, dll. Yang akan kita pakai adalah Visual Studio Code



Project 1

Change the Color

Hello Cobee!

Let's Change the Color!

Click Me!

This is yellow



Masih ingat tidak konsep dibawah ini?

1. Variable?
2. Array?
3. Function?

Kali ini kita akan membuat website yang bisa diganti-ganti backgroundnya!

<https://timedoctor-change-the-color.netlify.app/>

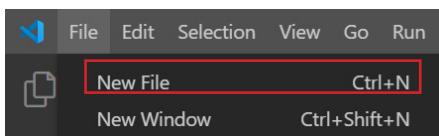
Tampilannya akan dibuat dengan **HTML**

Stylenya akan dibuat dengan **CSS**

Dan tentunya untuk menampilkan warna dan nama warna menggunakan button dibuat dengan **JavaScript!**

I. Preparation

1. **Buka** visual studio code
2. **Buat file** baru, pilih **File** lalu **New File**



3. **Save file** dengan format HTML.

Pilih **File** lalu klik **Save**

Beri nama project dengan : *Change the Color.html*





2. Build Web Structure

Pertama kita harus buat dulu struktur dari web dengan HTML. Tulis code ini pada file HTML yang sudah kita buat!

Codes

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>

  </body>

</html>
```

3. Run Program



Bagaimana cara menjalankan program?
Apakah kamu menemukan tombol Play?

Saat menggunakan Teks Editor, ada beberapa cara yang bisa digunakan untuk menjalankan program.



Coba jalankan programmu dengan cara dibawah ini ya!



1 -Membuka dari file HTML

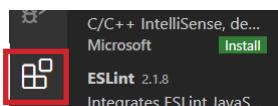
Kita dapat menjalankan program HTML dengan cara membuka file HTML yang telah disimpan dalam komputer.

1. Buka folder tempat menyimpan file HTML.
2. Buka file HTML tersebut, file akan terbuka langsung di web browsermu

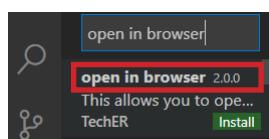
2 -Dengan VSCode Extension

Kedua, kita dapat menggunakan ekstensi Visual Studio Code untuk mempersingkat menjalankan program HTML.

1. Klik menu **Extension**



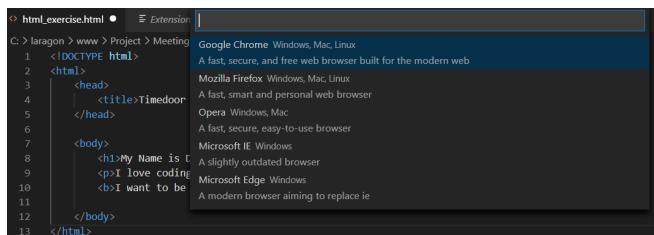
2. Pada search bar ketik **open in browser**. Lalu klik **Install**.



3. Setelah terinstal, kembali ke **dokumen HTML** yang dibuat. Untuk menjalankan program gunakan **shortcut** dibawah ini.

Alt + B	Untuk menjalankan program HTML di default browser.
Shift + Alt + B	Memilih browser tertentu untuk menjalankan program HTML.

Shift + Alt + B
Akan muncul pilihan browser untuk menjalankan program





Jangan takut jika browsernya kosong ya!
Itu terjadi karena kita baru membuat kerangkanya saja,
belum mengisi web dengan elemen apapun!

4. Give it Title

Sekarang kita tambahkan dulu judul untuk halaman web yang akan kita buat. Tambahkan code ini diantara tag

`<head> </head>`

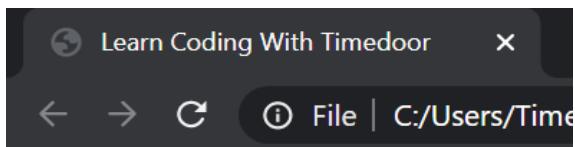


Codes

```
<title>  
    Learn Coding With Timedoctor  
</title>
```



Jangan lupa di Save kemudian coba reload halaman webmu!
Cek apakah di judul Tabnya sudah menjadi Learn Coding with Timedoctor?



5. The Body Content

Selanjutnya, kita buat konten nya, kita akan mengisi body halaman web dengan tulisan dan button untuk mengganti warna.

Pertemuan 1-How Cool is Programming World?

• CHANGE THE COLOR



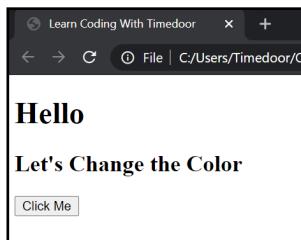
Tambahkan code ini diantara tag `<body> </body>`

Codes

```
<div class="center">
    <h1 id="greeting"> Hello </h1>
    <h2>Let's Change the Color</h2>
    <button> Click Me </button>
</div>
```



Jangan lupa di Save kemudian coba reload halaman webmu!
Cek apakah sudah terdapat tulisan dan button seperti ini?



6. The Style

Bukankah tampilannya terlalu biasa? Sekarang yuk kita berikan CSS agar tampilannya lebih menarik! Tulis CSS code ini diantara tag `<head> </head>`

Codes

```
<style>
    div {
        padding-top: 100px;
        text-align: center
    }
    button {
        background-color :red;
        color: white;
        text-align: center;
        font-size: 20px;
    }
</style>
```



Jangan lupa di Save kemudian coba reload halaman webmu!
Cek apakah tampilannya sekarang sudah berubah?

Hello

Let's Change the Color

Click Me



HTML dan CSS bisa dieksplor lagi untuk membuat tampilan yang menarik, tapi yang akan menjadi fokus kita adalah bagaimana web itu bekerja dengan menggunakan JavaScript.



Coba tekan buttonnya , apa yang terjadi?

Untuk membuat kejadian terjadi ketika button ditekan kita akan menambahkan JavaScript.

**Semua code JavaScriptnya akan ditulis di file HTML ini,
yang letakkan dalam tag <script>**



7. Change HTML Element

Sekarang kita akan mulai code JavaScriptnya.

Masih ingat tidak tentang variable?

Sebelum webnya dimuat, web akan menanyakan namamu terlebih dahulu, nama tersebut disimpan dalam variable dan valuenya akan ditampilkan ke dalam halaman web.

Letakkan code ini dibawah code </body>

Codes

```
<script>
    //Initializing variables
    const button = document.querySelector('button')
    const body = document.querySelector('body')

    var name=prompt("What's Your Name?")

    //Print variable value in HTML body
    document.getElementById("greeting").innerHTML
    = "Hello " + name + "!";

</script>
```



Menurutmu, apa fungsi variable button dan body?
Apa nilai dari variable name?

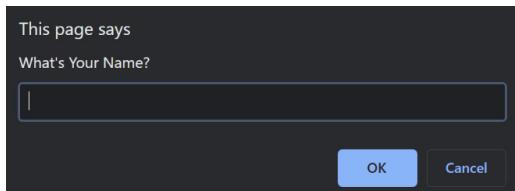


Code `document.getElementById` berfungsi untuk memilih elemen HTML berdasarkan id tertentu. Dalam program diatas, kita mengubah nilai yang ada pada id `greeting` menjadi `"Hello " + name + "!"`



Jangan lupa di Save kemudian coba reload halaman webmu!

Cek apakah program menampilkan prompt?



Coba isi prompt dengan namamu, lalu klik OK. Sekarang cek tampilan webnya, apakah berubah?



8. List of Colors



Masih ingat tentang Array? Array digunakan untuk menyimpan nama-nama warna.

Tuliskan code ini di dalam tag <script>...</script>



Codes

```
var colors = ['red', 'green', 'blue',
              'yellow', 'pink', 'purple', 'violet']

body.style.backgroundColor = colors[0]
```

• CHANGE THE COLOR



Jangan lupa di Save kemudian coba reload halaman webmu!



Apakah backgroundnya berubah warna?
Mengapa warna tersebut yang ditampilkan?



Code `body.style.backgroundColor` berfungsi untuk mengganti style body dengan warna background tertentu.

9. Function Change Color



Masih ingat tentang Function?

Sekarang kita buat function untuk mengganti warna

Tuliskan code ini masih di dalam tag

```
<script>...</script>
```



Codes

```
function changeColor(){
    // choose random number from array colors
    var colorIndex= parseInt(Math.random()*colors.length)
    // set background to color Index
    body.style.backgroundColor = colors[colorIndex]
}
```



Jangan lupa di Save kemudian coba reload halaman webmu!

Lalu coba tekan buttonnya, apakah warna backgroundnya dapat berganti?



Menurutmu, kenapa belum terganti?



10. When Button Clicked



Masih ingat apa itu event?

Kita akan membuat event ketika button diklik, maka function change color dipanggil.

Tuliskan code ini masih di dalam tag `<script>...<script>`



```
// when button clicked, call function changeColor  
button.addEventListener('click', changeColor)
```

Yay! Sekarang web nya sudah selesai!

Jangan lupa di Save kemudian coba reload halaman webmu! Sekarang coba klik buttonnya! Apakah programmu sudah berjalan dengan benar?



Challenge!

Hello Cobee!

Let's Change the Color

Click Me

This is pink

Perhatikan gambar yang beri tanda merah! Sekarang coba tambahkan agar setiap button diclick, terdapat tulisan yang menyatakan warna dari background yang terpilih!

Pertemuan 1-How Cool is Programming World? CHECK YOUR UNDERSTANDING!



Hari/Tanggal : _____

1. Menurutmu, apakah semasa remaja ini kamu juga bisa menjadi app/ game developer? ada ide apa yang akan kamu program?

.....
.....
.....

2. Apa saja elemen dalam sebuah halaman web? Sebutkan fungsinya!

.....
.....
.....

3. Jelaskan bagaimana JavaScript dalam project Change the Color?

.....
.....
.....

4. Tag apa yg digunakan untuk mengawali dan mengakhiri code javascript dalam html?

.....
.....
.....





2

NUMBER GENERATOR



Apa Yang Akan Kita Pelajari?

1. Pengenalan flowchart
2. Penggunaan For Loop dalam JavaScript
3. Pentingnya indentation dalam coding



Pertemuan

2

NUMBER GENERATOR WEB PAGE



Warm Up!



Menuliskan perintah yang sama berulang-ulang kali melelahkan bukan?

Masih ingat konsep Loop yang kita pelajari di Level Beginner? Ada berapa bentuk Loop yang kalian ketahui?

Hari ini kita akan kembali belajar tentang **Loop dalam bentuk For Loop** dan penggunaannya **dalam JavaScript**.



For Loop in JavaScript



Masih ingat apa itu **For Loop**?

Kapan For Loop digunakan? Bagaimana penulisan code for loop?

for counter from 1 to 10 count by 1

Syntax!

```
for (setup; condition; increment){  
    commands();  
}
```



Masih ingat fungsi **setup, condition, dan increment?**



Setup	Bagian setup digunakan untuk define variable untuk menampung nilai perhitungan berapa kali loop dijalankan. Contoh : i=0; berarti perulangan dimulai dengan nilai i sama dengan 0
Condition	Kondisi sampai mana loop akan berhenti/selesai . Contoh jika kondisinya i < 5, berarti perintah akan berhenti berulang jika nilai i kurang dari 5.
Incre- ment	Increment digunakan untuk mengubah nilai setup . Contoh ; i++ artinya menambah 1 setiap kali perulangan, i-- artinya mengurangi 1 setiap kali perulangan.

Quiz



Codes

```
for (var i=0; i<6; i++){
    console.log("I'm Cobee");
}
```

- Berapa nilai awal variabel i?

.....

- Apa kondisi stop perulangan diatas?

.....

- Berapa increment (penambahan nilai i) ?

.....

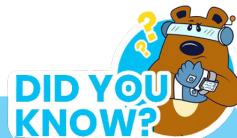
- Berapa kali command `console.log("I'm Cobee");` akan dijalankan?

.....



Indentation

 Apakah kamu menyadari bahwa kita menggunakan tab dalam menulis perintah yang ingin diulangi? Spasi tersebut disebut dengan **indentation atau white spaces**.



Tahukah kamu mengapa indentation penting dalam coding? Indentation digunakan membuat coding **terstruktur** sehingga **lebih mudah untuk membacanya**. Indentation mempermudah untuk melihat codes mana yang diulangi dan yang mana yang tidak.



Codes

```
// loop to print numbers 1 to 10
for (i = 1; i <= 10; i++) {
    // print number
    console.log(i);
}
// end of loop
```

Perhatikan code `console.log(i)`, code tersebut diberi indentation sebanyak 4 spasi (Tab), sedangkan tanda “`}`” tidak diberikan indentation.

Mengapa demikian?

Hanya code/perintah yang ingin diulangi diberikan indentation.



Quiz



Codes

```
for (i = 3; i >= 1; i--) {  
    console.log(i);  
}  
console.log("LIFTOFF!");
```



Menurutmu berapa kali perintah
`console.log("LIFTOFF!");` akan dijalankan?



DID YOU
KNOW?

Untuk mengetahui berapa kali sebuah perintah diulangi dalam for loops kita bisa menggunakan **Flowchart**.



Vocabulary

Flowchart

Suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses (algoritma) secara mendetail dan hubungan antara suatu proses dengan proses lainnya dalam suatu program.



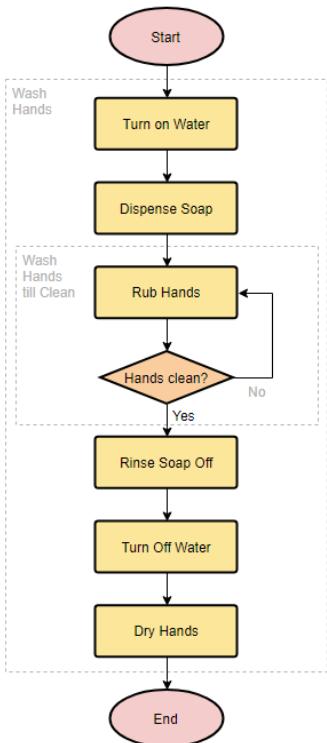
Flowchart

Sesuai dengan pengertian flowchart, yaitu bagan dengan simbol-simbol tertentu, maka kita harus tahu **simbol-simbol dasar** dalam flowchart.

Nama	Simbol	Fungsi
Terminator	Ellipsis	Simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan.
Process	Parallelogram	Proses atau pengolahaan yang dilakukan oleh komputer.
Data	Trapezoid	Input dan output data
Decision	Diamond	Pemilihan proses dari kondisi yang ada.
Flow	Arrow	Penghubung dari simbol satu ke simbol lainnya.



Yuk coba sekarang kita cermati flowchart dari algoritma mencuci tangan!



1. Mulai
2. Hidupkan air
3. Keluarkan sabun
4. Gosok tangan
5. Cek, apakah tangan sudah bersih?
6. Jika belum bersih, ulangi dari langkah 4
7. Jika sudah, bilas tangan
8. Matikan air
9. Selesai



Menurutmu, dari flowchart cuci tangan, **apakah ada kemungkinan perulangan?**



Mengapa penting untuk menggunakan flowchart?

Jika kamu kesulitan untuk memahami logika dalam sebuah program maka kamu bisa menggunakan **flowchart untuk membantu memahaminya!**
Selain itu, flowchart juga bisa membantu **memperjelas proses yang kompleks.**

• FLOWCHART



Codes

```
for (i = 0; i < 6; i++) {  
    console.log(i);  
}
```

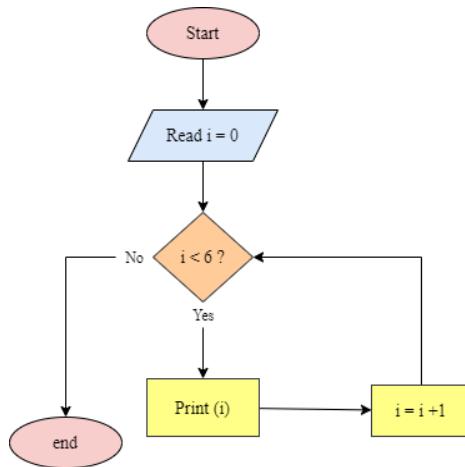


Gambar disamping merupakan flowchart dari code diatas.

Menurutmu berapa kali perintah cetak nilai i akan berulang?



Apakah dengan flowchart ini kamu lebih mudah untuk membaca dan memahami algoritma dari code diatas?





Project 1

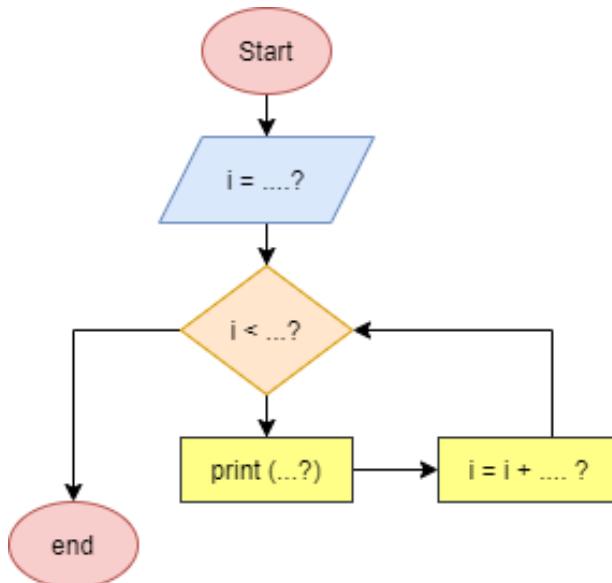
Number Sequence

Yuk kita coba buat web page yang menampilkan urutan bilangan menggunakan for loops JavaScript dan juga HTML!

Kali ini kita akan menampilkan 10 bilangan asli pertama.

Quiz

Bisakah kamu **melengkapi flowchart** untuk algoritma program yang **menampilkan 10 bilangan asli pertama**?



Pertemuan 2- Number Generator NUMBER SEQUENCE



1. Coding

Buka **visual studio code** lalu ketik code berikut ini.

Codes

```
<!DOCTYPE html>
<html>
    <head>
        <title>Learn For Loops</title>
    </head>

    <body>
        <b> 10 Natural Number Sequence<br></b>
        <script>
            for ( var i=0; i<10; i++){
                document.write(i);
                document.write("<br>");
            }
        </script>
    </body>
</html>
```

2. Save

Simpan code ke format HTML. Caranya klik **File → Save** Pilih folder yang kamu inginkan, beri nama "**html_js_1**" dan save as type "**HTML**".

3. Run Code

Sekarang coba jalankan program di browsermu dengan cara klik **Alt + B**. Apakah hasilnya sama seperti pada gambar berikut ini?



Learn For Loops

File | C:/laragon/www/urutan%20bilangan.html

10 Natural Number Sequence

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



Challenge!

Buat web yang bisa menampilkan **urutan bilangan genap dari 2 sampai 20** menggunakan For Loops JavaScript! Jika kesulitan, kamu bisa menggambarkan flowchartnya terlebih dulu!

Learn For Loops

File | C:/laragon/www/Project/Meeting%

Even Number

- 2
- 4
- 6
- 8
- 10
- 12
- 14
- 16
- 18
- 20

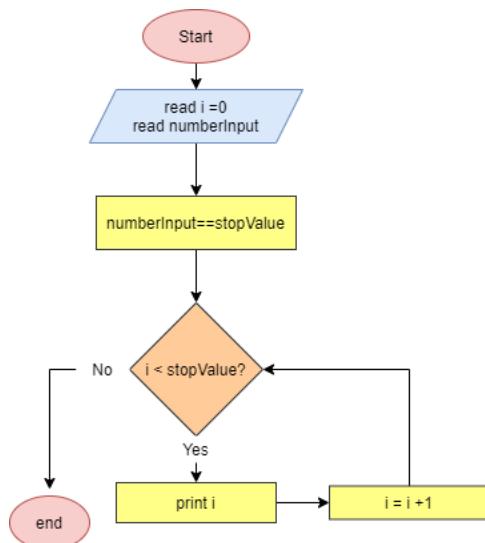


Project 2

Number Sequence by Input

Yuk kita coba buat web page untuk **generate urutan bilangan** sebanyak angka yang diminta Kali ini kita akan menggunakan **HTML Input form** dan juga **For Loops JavaScript!**

Sebelum membuat programnya, yuk **pahami dulu algoritmanya** berdasarkan flowchart disamping!





Kali ini kita hanya perlu **fokus pada coding JavaScriptnya** saja. HTML akan kita pelajari lebih lanjut!



Get Value form HTML Input

Nah input angka ditulis di web dalam dokumen HTML kan? Agar **angka itu bisa kita olah memakai Javascript** kita perlu input dengan `document.getElementById("....").value;`

Syntax!

```
document.getElementById("....").value;
```



id-nya diisi apa?

Tidak perlu bingung dahulu, kali ini kita **fokus belajar di JavaScriptnya** saja. Element id htmlnya bisa memakai template yg sudah disiapkan guru di **github**.



Ada yang tahu
apa itu GitHub?



GitHub merupakan aplikasi website yang digunakan untuk **menyimpan file berupa source code** (file coding). Selain itu, GitHub juga menjadi wadah untuk **kolaborasi dalam mengerjakan sebuah project**. Aplikasi ini sangat populer dikalangan programmer, bahkan Google dan Twitter juga menggunakaninya!



Pertemuan 2- Number Generator

NUMBER SEQUENCE

Yuk mulai kerjakan!

1. Prepare Template

1. Buka link di browser <https://github.com/cobee-codingbear/Meeting4-Project3>
2. Download source code dengan cara :

cobee-codingbear / Meeting4-Project3

Code Issues Pull requests Actions Projects Wiki Security Insights Setting

master 1 branch 0 tags

Go to file Add file Code

cobee-codingbear Add files via upload 9f1ae7a 3 minutes ago 3 commits

Project 3 Number Generator.html Add files via upload 3 minutes ago

Add a README

Klik tombol **Code** yang berwarna hijau.

Klik **Download ZIP** untuk mendownload file dalam bentuk ZIP.
Setelah selesai, **extract file** tersebut.
File akan berisi dokumen HTML.

Go to file Add file Code

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/cobee-codingbear/Me

Open with GitHub Desktop

Download ZIP

3. Buka **Visual Studio Code** lalu pilih **File** kemudian klik **Open File** dan pilih file HTML yang telah di download.

File Edit Selection View Go Run

New File Ctrl+N
New Window Ctrl+Shift+N

Open File... Ctrl+O
Open Folder... Ctrl+K Ctrl+O
Open Workspace...
Open Recent >



2. Fill the Code

Sekarang saatnya **melengkapi file HTML tersebut dengan JavaScript**. Jangan lupa, code JavaScript dituliskan dalam tag `<script>...</script>`

```
1  <!doctype html>
2  <html>
3  <head>
4  |   <title>Learn For Loops</title>
5  </head>
6  <body>
7  |   <b>Natural Number Generator</b>
8  |   <br>
9
10 |   <label for="input_stop">Enter Any Number </label>
11 |   <input id="input_stop" type="number">
12 |   <button onclick="sequence()">Submit</button>
13
14 </body>
15
16 <script>
17 |   // write your code here
18 </script>
19 </html>
```

Ketik code berikut ini.



Codes

```
<script>
  function sequence(){
    var stop=Number(document.getElementById("input_stop").value);

    for ( var i =0; i<stop; i++){
      document.write(i);
      document.write("<br>");
    }
  }
</script>
```



 artinya break yg digunakan untuk memutus line agar membuat line baru

2. Save

Simpan code ke format HTML. Caranya klik **File → Save** Pilih folder yang kamu inginkan, beri nama "**html_js_2**" dan save as type "HTML" → **Save**

3. Run Code

Sekarang coba jalankan program di browsermu!

Coba **masukkan angka berapapun** lalu tekan tombol **submit**. Apakah muncul barisan bilangan sebanyak angka yang kamu inputkan?



Challenge!

Buat web page yang menampilkan namamu sebanyak angka yang kamu inputkan! Gunakan code dibawah ini :
document.write("Namamu");



Hari/Tanggal :

1. Coba tuliskan code untuk mencetak angka 1 sampai 5 menggunakan for loops dalam JavaScript!

.....
.....
.....

2. `for (i = 10; i > 5; i++) {
}`

Berapa kali perulangan akan terjadi?

.....
.....
.....

3. Bagaimana flowchart dapat membantu dalam coding?

.....
.....
.....

4. Mengapa indentasi diperlukan dalam coding?

.....
.....
.....



3

EVENT ON QUIZ



Apa Yang Akan Kita Pelajari?

1. Event Listener dalam JavaScript
2. Membuat halaman web kuis dengan event listener



EVENT ON QUIZ



Warm Up!



Pernahkan kalian mengetik sebuah keyword di google search?
Apa yang terjadi setelahnya?



The screenshot shows a Google search results page for the query "Internet of things". The top result is a link to the Wikipedia article on the Internet of Things. The snippet from the page describes the Internet of Things as a network of physical objects ("things") embedded with sensors, software, and other technologies for the exchange of data over the Internet.

Setelah itu, maka akan muncul informasi terkait dengan keyword yang diketik sebelumnya.



Hal ini adalah salah satu contoh dari Input dan Output(I/O)
Ada yang tahu apa itu input dan output?



Input Output (I/O)



Vocabulary

Input

Semua data atau perintah yang dimasukkan oleh manusia ke komputer

Output

Informasi yang dikeluarkan oleh komputer dan bisa dibaca oleh manusia.



Kita memberi input ke google search berupa sebuah keyword. Lalu **Google Search memberi output** berupa informasi terkait dengan keyword.

Dalam dunia komputer tidak terlepas dari Input dan Output. Begitu juga dengan project web yang akan kita buat!



Event Listener



Masih ingat tentang event?

Coba ingat lagi project Change color!

Bagaimana program bisa tahu kalau kita sedang menekan tombol?

Coba kita cek kembali code yang dipakai ya!



Example

```
// when button clicked, call function changeColor
button.addEventListener('click', changeColor)
```

Dalam code tersebut kita menggunakan method `addEventListener` untuk mengecek apakah button diclick atau tidak!



Vocabulary

Event Listener

Event listener berfungsi untuk membuat event (click, change, mouseout, dll) untuk melakukan manipulasi terhadap elemen HTML

Pertemuan 3 – Event on Quiz

EVENT LISTENER



Perhatikan syntax event listener di JavaScript ya!

Syntax!

```
element.addEventListener(event, function, useCapture);
```

`element` adalah elemen HTML yang diberi event

Terdapat 3 parameter dalam Event Listener :

1. `event` adalah tipe event yang diberikan sesuai dengan HTML DOM.
Misal, “`click`”, “`mousedown`”, “`keydown`”, “`keypress`” dan lainnya
2. `function` adalah nama function yang dipanggil ketika event dijalankan.
3. `useCapture` adalah boolean dengan nilai True or False, parameter ini optional, bisa tidak digunakan.

Satu elemen bisa diberikan banyak Event

Satu elemen bisa diberikan tipe event yang sama dengan function yang berbeda.



Example

```
button.addEventListener('click', function1)
button.addEventListener('click', function2)
```

Satu elemen bisa diberikan tipe event yang berbeda



Example

```
button.addEventListener('click', function1)
button.addEventListener('keypressed', function2)
button.addEventListener('mousedown', function2)
```



Coding Practice!

Untuk lebih kenal dan lebih paham tentang event listener,
yuk kita kerjakan project dibawah ini!

Project 1

Event Keyboard

Event in Javascript



Game key:
w for move up
a for move left
s for move down
d for move right
space for shoot the enemy

Kali ini kita akan membuat simulasi ketika event pada keyboard menggunakan JavaScript!

timedoortest-event-keyboard.netlify.app

1. Preparation

Tampilannya akan dibuat menggunakan HTML, tapi kalian tidak perlu membuatnya!

Gunakan template yang sudah disediakan ya!

Download di : <https://github.com/timedoortestAcademy-smp/intl-3-eventKeyboard.git>

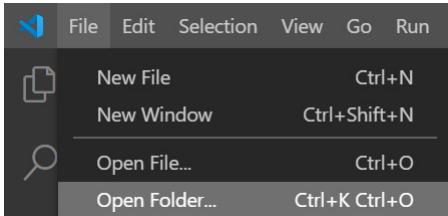
Pertemuan 3 – Event on Quiz

EVENT KEYBOARD



2. Open Template

Buka folder yang telah didownload tadi di visual studio Code!

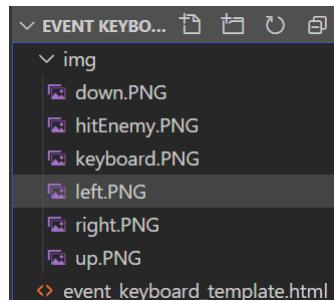


Buka folder yang telah didownload tadi di visual studio Code!

Dalam folder tersebut terdapat :

1. folder img, folder ini menyimpan gambar-gambar yang akan digunakan. Perhatikan perbedaan masing-masing gambar ya!

2. file HTML, di file HTML ini kamu akan menulis code nya!



Sekarang coba buka file **event_keyboard_template.html**

Lalu jalankan programnya!

Apa yang terjadi?



Coba berikan input dengan ketik huruf W/A/SD atau tekan spasi. Apakah sesuatu terjadi?

Event in Javascript

Type here



Game key:
w for move up
a for move left
s for move down
d for move right
space for shoot the enemy



3. Code

Agar programmu merespon ketika diberikan input, maka harus diberikan JavaScript code!

Ada 3 event yang akan dipakai yaitu :

1. **Key Down** : Ketika A/W/S/D ditekan
2. **Key Up** : ketika A/W/S/D tidak ditekan
3. **Key Press** : ketika space ditekan

Yuk kita mulai!

Semua code diketik dalam tag `<script>...</script>`

1. Initialize variable



Apakah kamu melihat
input form ?

Type here

Input form tersebut akan menjadi elemen yang diberikan event!

Ketik code dibawah ini!



Codes

```
var key_pressed = document.getElementById('field');
var keyid = document.getElementById('key');
```

Variable `key_pressed` menyimpan elemen input form.
Variable `keyid` menyimpan elemen gambar keyboard.

Pertemuan 3 – Event on Quiz

EVENT KEYBOARD



2. Event Listener

Elemen key pressed sudah dibuat! Sekarang saatnya menambahkan event listener ketika keydown, keyup, dan keypress. Tambahkan code dibawah ini!



Codes

```
key_pressed.addEventListener("keydown", onKeyDown);
key_pressed.addEventListener("keyup", onKeyUp);
key_pressed.addEventListener("keypress", onKeyPress);
```

Sekarang coba jalankan programmu!



Coba tekan tombol A/W/S/D atau Space bar.
Apakah sesuatu terjadi?

3. Define Function onKeyPress

Programmu belum mau berjalan, karena function nya belum di define!

Yuk mulai dengan define function ketika Space ditekan!



Codes

```
function onKeyPress(event) {
    if(event.code == "Space"){
        document.getElementById("status").innerHTML =
'Sprite shoot the enemy';
        keyid.src = 'img/hitEnemy.PNG';
    }
}
```



Bisakah kamu memahami program diatas?
Apa yang dijalankan dalam function tersebut?



4. Define Function onKeyDown

Selanjutnya tambahkan function ketika key ditekan. Dalam function ini kita akan menggunakan conditional.



Menurutmu, bagaimana conditional akan dipakai dalam function ini?

Yuk tambahkan code dibawah ini!



Codes

```
function onKeyDown(event) {
    if(event.key == "w" || event.key == "W"){
        document.getElementById("status").innerHTML =
'Sprite move up';
        keyid.src = 'img/up.PNG';
    }
    else if(event.key == "s" || event.key == "S"){
        document.getElementById("status").innerHTML =
'Sprite move down';
        keyid.src = 'img/down.PNG';
    }
    else if(event.key == "d" || event.key == "D"){
        document.getElementById("status").innerHTML =
'Sprite move right';
        keyid.src = 'img/right.PNG';
    }
    else if(event.key == "a" || event.key == "A"){
        document.getElementById("status").innerHTML =
'Sprite move left';
        keyid.src = 'img/left.PNG';
    }
    else {
        document.getElementById("status").innerHTML =
'Sprite doesnt move';
        keyid.src = 'img/keyboard.PNG';
    }
}
```



Coba jalankan programnya!

Apa yang terjadi ketika tombol A/W/S/D ditekan?
Mengapa terjadi seperti itu?

Pertemuan 3 – Event on Quiz

EVENT KEYBOARD



5. Define Function onKeyUp

Apakah gambar keyboard kembali kesemula ketika keyboard tidak ditekan?

Sekarang saatnya membuat function onKeyUp yang menjalankan :

Jika key w atau W, maka

Tampilkan tulisan *sprite stops move up*

Tampilkan gambar *keyboard.png*

Selain itu, jika s atau S, maka

Tampilkan tulisan *sprite stops move down*

Tampilkan gambar *keyboard.png*

Selain itu, jika a atau A, maka

Tampilkan tulisan *sprite stops move left*

Tampilkan gambar *keyboard.png*

Selain itu, jika d atau D, maka

Tampilkan tulisan *sprite stops move right*

Tampilkan gambar *keyboard.png*

Selain semua itu,

Tampilkan tulisan *sprite doesn't move*

Tampilkan gambar *keyboard.png*

Sekarang coba lengkapi code dibawah ini untuk membuat logika function diatas ya!



Do it Your Self !

```
function onKeyUp(event) {  
    if(event.key == "w" || event.key == "W"){  
        document.getElementById("status").innerHTML = 'Sprite  
stops move up';  
        keyid.src = 'img/keyboard.PNG';  
    }  
  
    // key s or S  
    // key a or A  
    // key d or D  
  
    // Else  
}
```



Programmu selesai!

Simpan dan Jalankan program.

Coba sekarang tekan tombol A/W/S/D/Space bar

Apa yang terjadi ketika kamu berhenti menekan tombolnya?



Ada berapa tipe event yang sudah kita gunakan?

Apa yang terjadi di tiap-tiap tipe event?

Untuk lebih kenal dan lebih paham tentang event listener,
yuk kita kerjakan project dibawah ini!

Pertemuan 3 – Event on Quiz

CHECK YOUR UNDERSTANDING



Hari/Tanggal: _____

1. Menurutmu, bagaimana input output dari program event keyboard?

.....
.....
.....

2. Bagaimana event listener digunakan dalam event keyboard ?

.....
.....
.....

3. Menurutmu, apa event type yang digunakan pada tombol subscribe di YouTube?



.....
.....
.....



Timedooring
Coding Academy



4

GUESSING GAME



Apa Yang Akan Kita Pelajari?

1. Implementasi conditional dalam JavaScript
2. Membuat halaman web guessing game



GUESSING GAME



Warm Up!

Guess The Number

We have selected a random number between 1 - 10. See if you can guess it.

Enter a guess:

Submit guess

Guesses Remaining: 10

Result :

Kali ini kita akan **membuat guessing game** menggunakan konsep **conditional** dan **function**.



Apakah kalian masih ingat?

Apa itu conditional dalam coding?
Apa itu function?

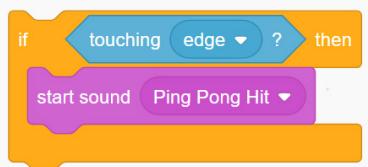


Conditional Logic

Yuk kita ingat-ingat lagi!



Masih ingat apa itu conditional?



Conditional digunakan untuk menjalankan perintah yang berbeda berdasarkan kondisi/syarat tertentu.

Dalam JavaScript, ada 3 conditional statement :

1. If Statement
2. Else Statement
3. Else if Statement

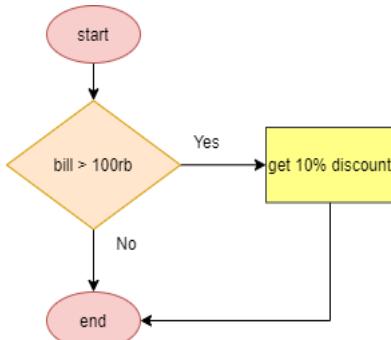


1. If Statement

If statement merupakan conditional yang hanya memiliki 1 kondisi untuk menjalankan perintah tertentu.



Coba perhatikan flowchart if statement dibawah ini!



Apa yang terjadi jika total belanja sebesar 150rb?

Apa yang terjadi jika total belanja sebesar 80rb?



Lalu bagaimana **penulisannya dalam JavaScript**?
masih ingat bagaimana sintaks if statement dalam JavaScript?

Syntax!

```
if (condition) {  
    //commands();  
}
```

Kondisi diletakkan di dalam tanda kurung (. . . .)
Sedangkan perintah yang akan dijalankan, diletakkan dalam kurung kurawal { . . . }

Flowchart diatas, jika dituliskan dalam **JavaScript** akan menjadi seperti coding dibawah ini!



Codes

```
if (bill > 100) {  
    document.write ("get 10% discount");  
}
```

Pertemuan 4- Guessing Game ELSE STATEMENT



2. Else Statement



Dalam contoh if statement diatas, bagaimana jika kita ingin **menjalankan perintah yang berbeda ketika kondisi tidak terpenuhi?**

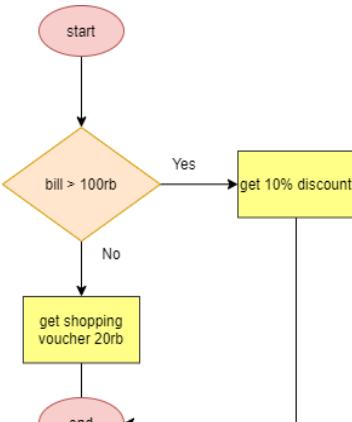
Misalnya, jika total belanja kurang dari 100rb, pelanggan akan mendapat voucher.



Untuk kondisi ini , kita bisa menggunakan **Else Statement!**

Else statement digunakan untuk menentukan perintah yang akan dijalankan jika kondisi bernilai salah.

Coba perhatikan flowchart else statement dibawah ini!



Dalam JavaScript, sintaks else statement dituliskan seperti ini.

Syntax!

```
if (condition) {
    //commands();
} else{
    // commands();
}
```



Codes

```
if (bill > 100) {
    document.write ("get 10% discount");
} else{
    document.write ("get shopping voucher20k");
}
```



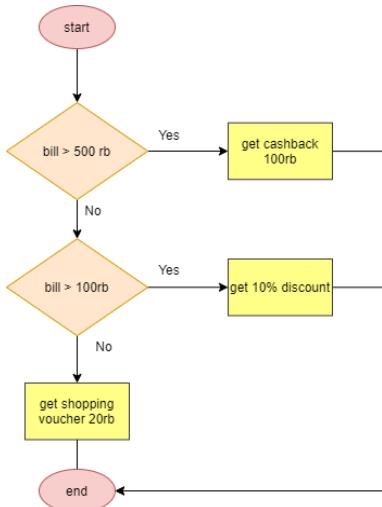
2. Else If Statement



If statement dan else statement hanya memiliki satu kondisi. Bagaimana jika kita ingin membuat banyak kondisi dalam program kita? Untuk kondisi ini , kita bisa menggunakan Else if Statement!

Else if statement merupakan conditional yang memiliki lebih dari satu kondisi.

Coba perhatikan **flowchart else if statement** dibawah ini!



Dalam **JavaScript**, sintaks else statement dituliskan seperti ini.

Syntax!

```

if (condition) {
  //commands();
} else if (condition) {
  // commands();
} else{
  // commands();
}
  
```



Codes

```

if (bill > 500) {
  document.write ("get cashback 100rb");
} else if (bill > 100) {
  document.write ("get 10% discount");
} else{
  document.write ("get shopping voucher 20k");
}
  
```



Quiz

Letter grade	Percentage
A	80 - 100
B	65 - 79
C	55 - 64
D	50 - 54
E	0 - 49



Greg mendapat nilai A dalam ujian coding. Dalam ujian tersebut, grade letter yang akan di dapat ditentukan dari nilai yang diperoleh dalam ujian.

Program dibawah ini adalah **program untuk menentukan letter grade**. Bisakah kamu melengkapi code dibawah ini?



Codes

```
if (score > .....){  
    document.write ("A");  
}else if (.....){  
    document.write ("B");  
}..... (score >54){  
    document.write ("....");  
}else if (.....){  
    document.write ("D");  
}.....{  
    document.write ("....");  
}
```



Jika Greg mendapat nilai 75, apa letter grade yang didapat?

Jika Santi mendapat nilai 52, apa letter grade yang didapat?



Coding Practice!

Guessing Game

<https://timedoors-guessinggame.netlify.app>

Guess The Number

We have selected a random number between 1 - 10. See if you can guess it.

Enter a guess:

Submit guess

Guesses Remaining: 10

Result :

Kali ini kita akan membuat sebuah **guessing game** menggunakan **conditional** yang telah kita pelajari! Dalam game ini, pemain harus menebak angka yang terpilih antara 1 sampai 10. Pemain memiliki 5 kesempatan untuk bermain.

**Coba lihat game jadinya yang ditunjukan guru ya!
Lalu coba bayangkan bagaimana algoritma dari
program tersebut!**

Main Project

Guessing Game

Yuk mulai kerjakan!

1. Prepare Template

1. Buka link di browser https://github.com/cobee-codingbear/guessGame_js
2. Buka file HTML di **Visual Studio Code**.



Pertemuan4- Guessing Game

GUESSING GAME ALGORITHM

2. Fill the Code

Sekarang saatnya **melengkapi file HTML** tersebut dengan JavaScript. Jangan lupa, **code JavaScript dituliskan dalam tag <script>...</script>**

Jika kalian lihat, sudah ada beberapa coding. Kita hanya perlu melengkapi kode pada **komentar nomor 1-5.**

```
97 <script type = "text/javascript">
98     const numberInputed = document.querySelector('#guessField');
99     const guessSlot = document.querySelector('.lastResult');
100    const resultCheck = document.querySelector('.resultCheck');
101    const p = document.createElement('p');
102
103    //1. Set Variable
104    document.getElementById("submitguess").onclick = function(){
105        // 2. Event when Guess Submited
106        //write the code here
107    }
108
109    function validateGuess(guess){
110
111        //3. Function to validate the guess
112        //write the code here
113    }
114    function checkGuess(guess){
115        //4. Function to check the Guess
116        //write the code here
117    }
118
119    function endGame(){
120        //5. Function to end the Game
121        //write the code here
122    }
123
124    function displayMessage(message){
125        resultCheck.innerHTML = `<b>${message}</b>`;
126        guessSlot.innerHTML = `${guessCount}`;
127    }
128
129 </script>
```



1 Set Variable

Pertama atur dulu **nilai variable** randomNumber dan guessCount.



Codes

```
var randomNumber = parseInt((Math.random()*10)+1);  
var guessCount = 5;
```

Variable randomNumber menampung nilai acak antara 1 sampai 10
Variable guessCount menampung nilai kesempatan menebak angka.

2 Event Guess Submitted

Saat button submit diklik, apa yang akan terjadi?



Codes

```
var guess = numberInputed.value;  
validateGuess(guess);
```

Ketika button submit diklik, angka yang dimasukkan akan disimpan di variabel **guess**. Setelah itu function **validateGuess** akan dijalankan. Namun, kita belum memiliki function tersebut. Yuk kita define function tersebut!

Pertemuan 4- Guessing Game

GUESSING GAME



3 Function Validate Guess

Function ini bertugas untuk **melakukan validasi**, apakah pemain memasukkan angka antara 1-10 dan apakah pemain masih punya kesempatan bermain.

Tuliskan codes berikut di dalam **function validateGuess(guess)**



Codes

```
if (guess<1){  
    alert('Please enter a number greater than 0!');  
}else if (guess > 10){  
    alert('Please enter a number less than 11!');  
  
}else if (guessCount === 0){  
    displayMessage("Game Over!!!");  
    endGame();  
}  
else{  
    checkGuess(guess);  
}
```

Ketika angka yang dimasukkan sudah sesuai (antara 1-10) dan masih memiliki kesempatan bermain, maka **function checkGuess dengan parameter guess** akan dipanggil. Value dari parameter guess ini akan diolah di function validateGuess



4 Function Check Guess

Function Check Guess bertugas untuk **mengecek nilai guess** apakah sama dengan nomor yang diacak atau tidak.

Tuliskan codes berikut di dalam function **checkGuess(guess)**



Codes

```
if(guess == randomNumber){  
    displayMessage("CONGRATULATIONS!!! You Guessed it right");  
    endGame();  
}  
else if(guess>randomNumber ){  
    guessCount--;  
    displayMessage("Oops! Try a smaller number!");  
}  
else if (guess<randomNumber){  
    guessCount--;  
    displayMessage("Oops! Try a greater number!");  
}
```

5 Function End Game

Function End Game bertugas untuk **menonaktifkan input dan button** agar tidak bisa digunakan lagi.

Tuliskan codes berikut di dalam **function endGame()**



Codes

```
numberInputed.value= " "; //empty input form  
//disable button  
document.getElementById("submitguess").setAttribute('disabled',  
'');
```



Pertemuan 4- Guessing Game

GUESSING GAME

3. Save

Simpan code ke format HTML dengan nama “**guessing-game**” dan save as type “HTML”

4. Run Code

Sekarang coba **jalankan program di browsermu** dan coba test apakah programmu sudah berjalan dengan benar?

Game Time!!

Coba mainkan gamemu dan berlomba dengan teman. **Siapa yang bisa menebak paling cepat?**



Challenge!

Coba modifikasi gamenya menjadi menebak angka dari 10-15.

Berikan validasi agar pemain hanya menginputkan angka 10 sampai 15.

Pemain memiliki 3 kesempatan, jika kesempatan habis maka game Over!

Pertemuan 4- Guessing Game

CHECK YOUR UNDERSTANDING



Hari/Tanggal : _____

1. Bagaimana conditional digunakan dalam guessing game?

.....
.....
.....

2. Berdasarkan pemahamanmu, bagaimana function parameter digunakan dalam guessing game?

.....
.....
.....

- 3.
- ```
if(guess>randomNumber) {
 guessCount--;
 displayMessage("Oops! Try a smaller number!");
}
```

Apa fungsi `guessCount--`; dalam conditional diatas?

.....  
.....  
.....



# 5

## CONDITIONAL LOOP IN CODE COMBAT



### Apa Yang Akan Kita Pelajari?

1. Conditional loop dalam JavaScript
2. Implementasi Condition Loop dalam Code Combat



Pertemuan

5

## CONDITIONAL LOOP IN CODE COMBAT



### Warm Up!

 Masih ingat tidak? Dalam coding, perintah apa saja yang bisa digunakan untuk **menjalankan perintah berulang-ulang**?

Sebelumnya kita sudah mempelajari **For Loop** untuk mengulangi perintah berulang-ulang. Sekarang mari kita pelajari bentuk loop lainnya yaitu **Conditional loop**.



Masih ingat tidak **apa itu conditional loop** dalam coding?  
Ada berapa bentuk conditional loop?



### While Loop

```
while [path ahead ?]
do []
```



Pernah melihat block ini?  
Apa fungsinya?



### Vocabulary

#### While Loop

While loop menjalankan kode/perintah yang sama berulang kali selama kondisi yang ditentukan benar.



## Syntax!

```
while (condition){
 //commands
}
```

**Condition** menentukan berapa kali perulangan akan dijalankan. Contohnya, kamu bisa mengisi `i < 10` sebagai kondisi, ini artinya, perulangan akan tetap dijalankan jika nilai i kurang dari 10.



### Codes

```
var i = 0;
while (i < 10) {
 console.log(i);
 i++;
}
```



Apa kondisi dari perulangan while loop disebelah?

## Using a Function in A While Loop

**Function** dapat digunakan sebagai condition dalam while loop. Perhatikan contoh dibawah ini!



### Codes

```
while (hasPathAhead()) {
 forward();
}
```



Berapa kali `forward()` akan dijalankan?



## Do While Loop

**Do While loop** adalah bentuk lain dari conditional loop.



### Vocabulary

#### Do While Loop

Do while loop akan menjalankan perintah terlebih dahulu kemudian mengecek kondisi perulangannya. Jika kondisi belum terpenuhi, perintah akan dijalankan kembali .

#### Syntax!

```
do{
 //commands
}while (condition)
```

Perhatikan contoh dibawah ini!



### Codes

```
do{
 forward();
}while(hasPathAhead())
```

Dengan do while perintah `forward();` akan dijalankan terlebih dahulu kemudian di cek apakah kondisinya benar atau tidak.

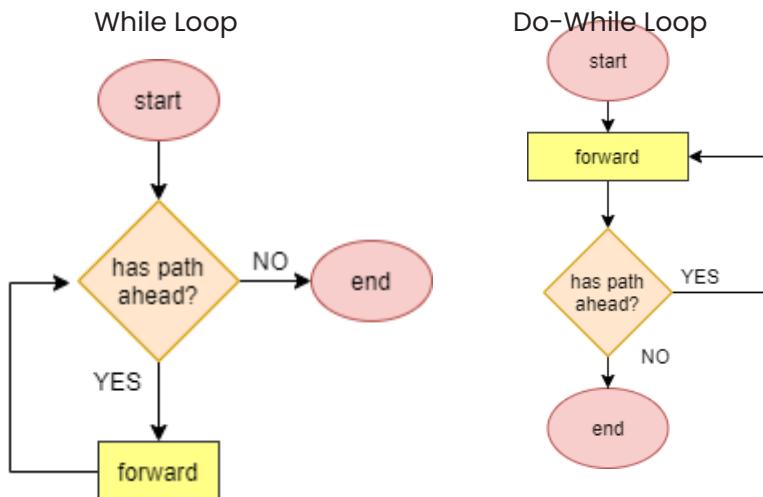


## While Loop vs Do While Loop

Setelah mengetahui **while loop** dan **do-while loop** dalam JavaScript, bisakah kalian mencari tahu perbedaan dari kedua bentuk conditional loop tersebut?



Coba lihat **flowchart** dari while loop dan do while loop dibawah ini dan **temukan perbedaannya!**



Perulangan do/while akan mengecek kondisi di belakang (sesudah mengulang), sedangkan while akan mencek kondisi di depan atau awal (sebelum mengulang).

## CODING PRACTICE



### Coding Practice!

## Code Combat



Dalam game ini, kita akan menyelesaikan misi perang menggunakan conditional loop yang sudah kita pelajari!

## Preparation!

1. Buka url di browsermu : [codecombat.com](https://www.codecombat.com)
2. Kemudian **LOGIN** menggunakan akun timedoctor coding academy.
3. Setelah itu, klik **Continue Playing**
4. Maikan **Level Kithgard Dungeon**





## Ayo kita selesaikan tantangannya!

Mainkan semua level!

Dimulai dari pintu masuk sampai pintu keluar!

Jangan lupa untuk ubah bahasanya ke JavaScript ya!



### CHECK YOUR UNDERSTANDING



Hari/Tanggal :

1.

```
var counter = 0;
while (counter < 3) {
 forward();
 counter++
}
```

Dari while loop diatas, berapa kali forward akan dijalankan?

.....  
.....  
.....

2. Mengapa dalam menyelesaikan misi Code Combat diperlukan conditional loop?

.....  
.....  
.....

3. Apa perbedaan while loop dengan do while loop?

.....  
.....  
.....



**Timedoctor**  
Coding Academy



# 6

# INTERNET RETURN VALUE



## Apa Yang Akan Kita Pelajari?

1. Bagaimana cara kerja internet
2. Penggunaan internet sebagai sumber belajar
3. Pemahaman lebih lanjut tentang function return value



## INTERNET RETURN VALUE



### Warm Up!

Dalam kehidupan sehari-hari, hampir tidak pernah lepas dengan **Internet**.



Menurutmu, sebenarnya **internet itu apa?**

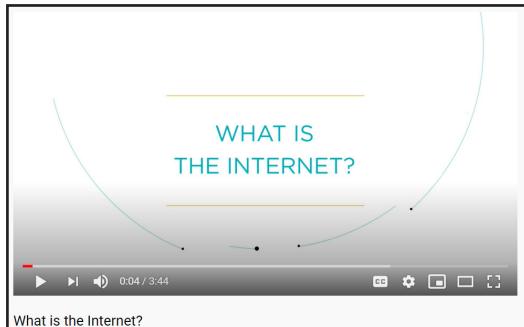
Sesungguhnya, siapa yang memiliki internet?



### What is Internet?



### Check This Video



### Vocabulary

#### Internet

Internet merupakan jaringan terluas dalam sistem teknologi informasi yang memungkinkan perangkat di seluruh dunia untuk saling terhubung.



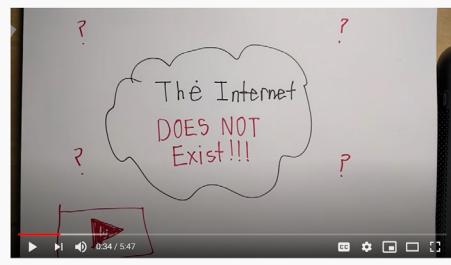
## How Does Internet Works?



Dengan internet, **sangat mudah untuk mencari informasi** hanya dengan mengetikkan keyword tertentu. Mengapa bisa demikian? Apakah kamu tahu **bagaimana cara kerja internet?**



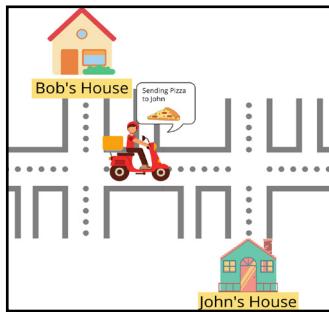
### Check This Video



How Does The Internet Work?

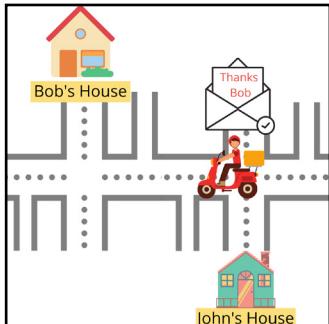
**Secara singkat**, internet bekerja seperti cerita dibawah ini :

1



**Bob dan John** tinggal di kota yang sama, rumah mereka **berjarak 50 km**. Suatu hari, Bob ingin **mengirim Pizza ke rumah John**, tetapi dia tidak memiliki kendaraan untuk mengirim Pizza maka dari itu John **perlu kurir** untuk mengambil dan mengirimkan Pizza ke John

2



**John merasa senang**, sehingga ia **mengirimkan surat melalui kurir** sebagai bentuk terima kasih kepada Bob

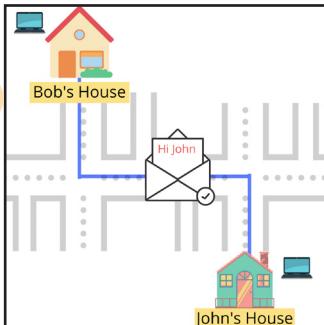


## Pertemuan 6- Internet Return Value

### HOW DOES THE INTERNET WORKS?

Sekarang mari simak bagaimana sebenarnya internet bekerja!

1



Sekarang **Bob dan John sudah memiliki komputer** di rumahnya masing-masing. Tidak hanya jalan, sekarang **komputer di rumah Bob dan John terhubung** sehingga Bob bisa berkirim pesan ke John melalui kabel.

Hal ini disebut dengan **Network**.



#### Vocabulary

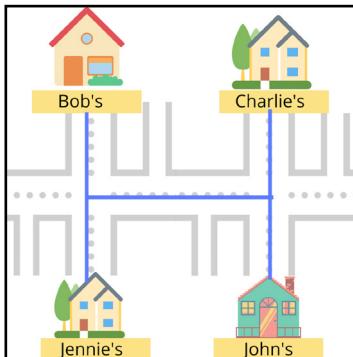
#### Network

Sekelompok komputer yang terhubung yang bisa saling berbagi sumber daya dan data.

Jika terdapat lebih dari 1 network yang saling terhubung itu disebut dengan **Interconnected network (Internet)**



**Bagaimana jika terdapat banyak komputer** dalam sebuah jaringan?  
Bagaimana cara mengirim pesan dari komputer 1 ke komputer 2 **agar tidak salah alamat?**

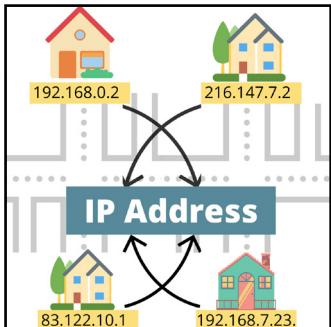


## Pertemuan 6- Internet Return Value HOW DOES THE INTERNET WORKS?



Sama seperti rumah, komputer memerlukan sebuah alamat yang dikenal dengan **IP Adress**

2

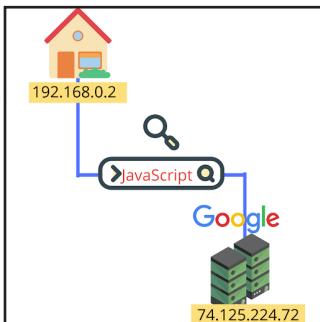


### Vocabulary

#### IP Adress

Nomor biner atau identitas numerik yang dipakai disetiap komputer agar komputer tersebut dapat saling berhubungan

3



Ketika kamu ingin mencari informasi di Google, **server mu harus terhubung ke server Google** (komputer harus mengetahui alamat IP Google) dan **Google akan mengirimkan informasi kembali** melalui website.



Apakah kita harus **menghafal semua alamt IP?**  
Tentunya tidak karena ada bantuan **DNS**



**DNS (Domain Name Server)** berfungsi untuk **menerjemahkan alamat IP Address menjadi kata-kata** dari situs tertentu sehingga kita bisa mengakses situs tersebut dengan mudah (tanpa harus menghafal alamat IPnya).

www.example.com → 192.168.24.5



## Function

### Yuk kita ingat-ingat lagi!

-  Masih ingat apa itu **function** dalam coding?  
Bisa berikan contoh function di kehidupan sehari-hari?

**Function** merupakan perintah yang mewakili sekumpulan program untuk melakukan tugas tertentu.



Bagaimana function digunakan dalam JavaScript ya?

### Syntax!

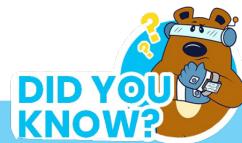
```
function function_name(){
 //lines of code to be executed
}
```

Dalam penulisan function, selalu diawali dengan keyword **function** lalu diikuti dengan nama function, kemudian diikuti kurung kurawal { . . . }.

Isi dari function dituliskan dalam { . . . }

-  Masih ingat tidak bagaimana **pemberian nama function** yang baik? Lalu **setelah function** dibuat, apakah function akan langsung dijalankan oleh program?

```
1 //move through the sharp corners.
2
3 forward();
4 turnLeft();
5 forward();
6 turnRight();
7
```



**Pemanggilan function** ini sering kita lakukan saat bermain Tynker JavaScript ataupun CodeCombat!  
Ada yang tau bagaimana caranya?

## Syntax!

```
function_name();
```

Pemanggilan function dilakukan dengan menulis nama function diikuti dengan tanda kurung () dan semicolon ;

Perhatikan contoh function berikut ini.



### Codes

```
//define a function
function myFunction(){
 document.write("This is a simple function");
}

for (i=0; i<10; i++){
 myFunction(); //function call
}
```



Apa yang akan dijalankan dari function diatas?

Berapa kali kalimat **This is a simple function** akan dipanggil?



### Function Parameter



Sebuah function dapat memiliki parameter di dalamnya.  
Masih ingat tidak apa itu parameter?

Sebuah function dengan  
parameter

**Luas = Panjang x Lebar**

Panjang dan lebar adalah  
parameter

Parameter merupakan nilai/ukuran yang ada dalam sebuah function.

### Syntax!

```
function function_name(parameter){
 //lines of code to be executed
}
```

Pemanggilan function dengan parameter dilakukan dengan menulis nama function diikuti dengan parameter yang diletakkan dalam tanda kurung (), lalu semicolon ;

### Syntax!

```
function_name(parameter);
```



Perhatikan contoh function berikut ini.



### Codes

```
//define a function
function area(length, width){
 return length * width;
}

//function call
area(10, 6);
```



Apa yang akan dihasilkan dari pemanggilan function  
`area(10, 6);`?



### Codes

```
function greeting(name, age){
 if(age > 10){
 return "Hello my name is " + name +
 ". I am " + age + " years old";
 } else {
 return "Hello, you can call me " + name;
 }
}
console.log(greeting("Kevin", 20));
console.log(greeting("Cobee", 2));
```



Apa yang akan dihasilkan pada console?

Jika kamu perhatikan terdapat keyword `return` pada setiap contoh code di atas. Tahuhan kamu apa fungsi `return`?

### RETURN KEYWORD

#### Return Keyword



#### Vocabulary

##### Return

Sebuah statement dalam program yang akan menghentikan eksekusi function dan mengembalikan nilai dari sebuah function.

Terdapat dua jenis statement return pada pemrograman, yaitu statement return tanpa pengembalian nilai dan statement return dengan nilai yang spesifik

#### Syntax!

```
return; //return without a value
//return with a value
return true;
return false;
return a + b;
```

Agar lebih paham, kita perhatikan contoh code dibawah ini ya.



#### Codes

```
function counter(){
 for(var count = 0; ; count++){
 console.log(count);
 if(count === 10){
 return;
 }
 }
}
counter();
```



Apa yang dihasilkan dari pemanggilan function `counter()`? Apakah fungsi statement return pada code tersebut?

**Yuk simak contoh lain.**



### Codes

```
function magic() {
 return function calc(x) {
 return x * 42;
 };
}

var answer = magic();
console.log(answer(1337));
```



Apa yang dihasilkan di console? Bisakah kamu jelaskan bagaimana fungsi return dalam code di atas?

Sebuah function yang memiliki return didalamnya disebut juga sebagai function return.



### Vocabulary

#### Function Return

Sebuah function yang memiliki statement return, sehingga saat eksekusi dihentikan function memiliki nilai akhir..



Sekarang, bisakah kamu membuat sebuah function return? Coba buat satu dan tunjukkan ke gurumu!

# Pertemuan 6- Internet Return Value

## THERMOMETER CONVERTER

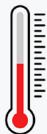


### Project

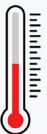
#### Thermometer Converter

Thermometer Converter

Click to convert



Termometer Celcius



Termometer Fahrenheit



30 derajat celcius  
berapa fahrenheit ya?  
Pernah tidak kamu  
diminta untuk  
mengkonversi suhu?

<https://timedoortemperature-converter.netlify.app/>

Kali ini kita akan membuat web yang bisa mengkonversi celcius ke fahrenheit dan sebaliknya.

### 1. Preparation

Gunakan template yang sudah disediakan ya!

Download di : <https://github.com/timedooraAcademy-smp/intl-6-temperature-convert>

Buka file HTML di Visual Studio Code!

Coba jalankan programnya!

Lalu input angka di formnya,  
kemudian klik convert.

```
MEETING 5 - THERMOMETER.html X
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Timedoort Temperature Converter</title>
6 </head>
7
8 <style>
9 body, html {
10 width: 100%;
11 height: 100%;
12 margin: 0;
13 }
14
15 .container {
16 width: 100%;
17 height: 100%;
18 }
19
20 .toppane{
21 position: relative;
22 width: 100%;
23 height: 100px;
24 border-collapse: collapse;
25 background-color: #white;
26 }
27
```



Dapatkah program melakukan konversi?



## 2. Codes

Program belum bisa berjalan dengan baik. Karena code JavaScriptnya belum ditambahkan.

### **Yuk kita mulai!**

Semua code diketik dalam tag `<script>...</script>`

### **1. Function Convert Temperature.**

Pertama, buat function yang bernama `convertTemp()` yang berfungsi untuk melakukan konversi temperature.



#### **Codes**

```
function convertTemp(){
 var celcius = document.getElementById("Celcius");
 var fahrenheit = document.getElementById("Fahrenheit");

 var celc = parseFloat(celcius.value);
 var fahr = parseFloat(fahrenheit.value);

 var convert;
 //conditionals

}
```

Dalam function tersebut, kita define beberapa variable :

1. Variable celcius berisi elemen form input celcius
2. Variable fahrenheit berisi elemen form input fahrenheit.
3. Variable celc berisi nilai celcius berbentuk desimal dari input yang diberikan.
4. Variable fahr berisi nilai fahrenheit berbentuk desimal dari input yang diberikan.
5. Variable convert digunakan untuk menampung nilai konversinya.



## Pertemuan 6- Internet Return Value THERMOMETER CONVERTER

### 2. The Conditionals

Dalam function `convertTemp()` kita juga perlu memberikan kondisi. Tulis code ini dibawah komentar `//conditionals`



```
if(isNaN(celc) && isNaN(fahr)){
 return alert('Fill with number');
}
else if(isNaN(celc) || isNaN(fahr)){
 if(isNaN(celc)){
 convert = (5 * (fahr - 32)) / 9;
 return celcius.value = convert;
 }
 else if(isNaN(fahr)){
 convert = ((9 * celc) / 5) + 32;
 return fahrenheit.value = convert;
 }
}
else {
 alert('Fill in one form!');
 celcius.value = "";
 fahrenheit.value = "";
 return;
}
```



Dapatkah kalian memahami coding diatas?



Ingatkah kamu rumus konversi celcius ke fahrenheit atau sebaliknya?



Jika variable celc dan fahr bukan angka, maka  
return alert "Fill with number"

Jika variable celc atau fahr bukan angka

    Jika variable celc bukan angka/kosong

        maka dihitung nilai celc dan dikembalikan

    Jika variable fahr bukan angka/kosong

        maka dihitung nilai fahr dan dikembalikan

Selain itu

    Akan di alert "fill in one form"

    nilai celcius dan fahrenheit dikosongkan

    kemudian nilainya dikembalikan.



## Pertemuan 6- Internet Return Value

### CHECK YOUR UNDERSTANDING

Hari/Tanggal : \_\_\_\_\_

- 1.** Bagaimana cara kerja internet?  
Mengapa kita perlu membayar internet?

.....  
.....  
.....

- 2.** Berdasarkan pemahamanmu, apa fungsi return dalam javascript?

.....  
.....  
.....

- 3.**

```
var a=10;
var b=5;

function myFunction(){
 c= a*b;
 return c;
}
myFunction();
```

Apa output dari kode program diatas?

.....  
.....  
.....



**Timedooring**  
Coding Academy



# 7

## JAVASCRIPT HEAP AND STACK



### Apa Yang Akan Kita Pelajari?

1. Bagaimana JavaScript mengeksekusi program
2. Implementasi JavaScript dalam Catch the Bird Project



## JavaScript Heap and Stack



### Warm Up!

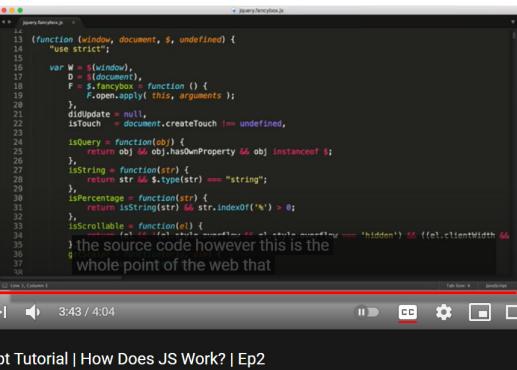


Program web apa saja yang sudah pernah kamu buat dengan JavaScript?

Setelah mencoba beberapa project. Sudahkah kamu paham **bagaimana sebenarnya JavaScript bekerja?**



### Check This Video



```
13 (function (window, document, $, undefined) {
14 "use strict";
15
16 var W = ((window),
17 D = (document),
18 $ = (jQuery));
19
20 F.open = function () {
21 F.open.apply(this, arguments);
22 };
23 didUpdate = null,
24 isTouch = document.createTouch ? undefined,
25 isQuery = function (obj) {
26 return obj && obj.hasOwnProperty(&& obj instanceof);
27 };
28 isString = function (str) {
29 return str && $.type(str) === "string";
30 };
31 isPercentage = function (str) {
32 return isString(str) && str.indexOf("%") > 0;
33 };
34 isScrolable = function (el) {
35 return (el.offsetWidth && el.offsetHeight && el.style.visibility === 'hidden') && (el.clientWidth &&
36 el.clientHeight) || !el.scrollHeight;
37 };
38
39 // the source code however this is the
40 // whole point of the web that
41 }
```

The video player interface includes a play button, volume control, and a progress bar showing 3:43 / 4:04.

Javascript Tutorial | How Does JS Work? | Ep2



Setelah menonton video tersebut, apakah kamu paham bagaimana cara kerja JavaScript?



Browser yang kita pakai punya mesin khusus untuk membaca kode Javascript atau yang biasa disebut **Javascript Engine**. Ada berbagai macam javascript engine, contohnya V8 yang digunakan oleh Google.



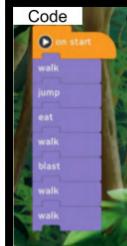
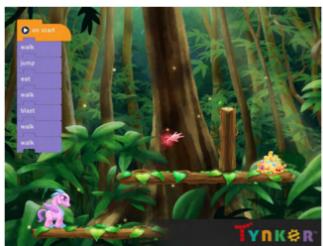
## JavaScript Engine



### Vocabulary

#### JavaScript Engine

Mesin yang dapat membaca dan mengolah kode-kode Javascript agar dapat di eksekusi oleh komputer. Lalu hasil eksekusi kode javascript akan ditampilkan di browser atau di console.

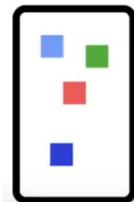


Masih ingat bagaimana cara memainkan game Dragon Blast diatas?

Bagaimana **jika susunan kodennya diputar balik?**

JS

Memory Heap



Call Stack



**Javascript Engine** dalam mengeksekusi kode-kode javascript terdiri dari dua bagian yaitu **Memory Heap** dan **Call Stack**.

### MEMORY HEAP



### Memory Heap



Masih ingat apa itu **variable global** dan **variable lokal** yang pernah kita pelajari di kelas Beginner?

#### Variable Global

```
var score = 0;
```

#### Variable Lokal

```
function funct(){
 var x = 20;
}
```

VS



Coba jelaskan perbedaan di atas ya!

Variable global akan disimpan di **memory heap**.



#### Vocabulary

#### Memory Heap

Memori yang terdapat didalam JavaScript engine yang dapat menyimpan data dalam urutan acak. Data yang akan bertahan untuk waktu yang cukup lama (variable global) akan masuk kedalam heap.



Setelah seluruh program selesai dilaksanakan, maka data-data dalam heap akan dihapus.



## Call Stack



### Vocabulary

#### Call Stack

Sebuah penyimpanan data yang bersifat **LIFO (Last In, First Out)**.

Untuk tahu cara kerja stack, coba perhatikan code dibawah ini.



## Codes

```
function baz(){
 console.log("Hello from cobee!");
}

function bar(){
 baz();
}

function foo(){
 bar();
}

foo();
```



Apa yang akan dimunculkan console saat program dijalankan? Bagaimana urutan eksekusi programnya?

## Pertemuan 7- JavaScript Heap and Stack

### CATCH THE BIRD



**Yuk kita pahami algoritma code tersebut dari gambar dibawah!**

#### JavaScript Program

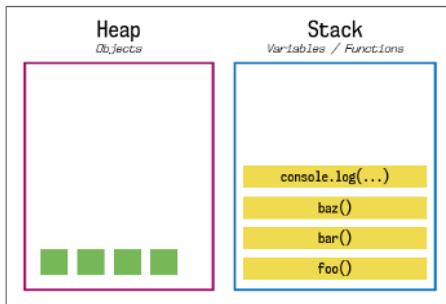
```
function baz() {
 console.log('Hello From baz');
}

function bar() {
 baz();
}

function foo() {
 bar();
}

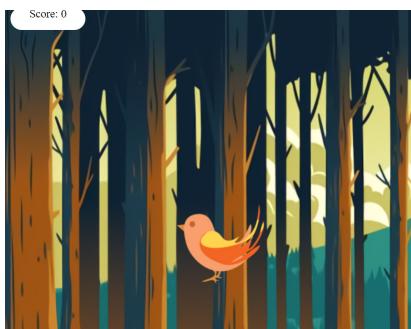
foo();
```

#### JavaScript Runtime



## Project

### Catch the Bird



Project yang akan dibuat adalah **catch the bird!**

Dalam project ini, kita punya 10 detik untuk menangkap burung dengan cara klik pada burung!

Jika burung yang berhasil ditangkap masih kurang dari 5, maka kamu akan kalah.

<https://timedoctor-catchthebird.netlify.app/>



## 1. Preparation

Gunakan template yang sudah disediakan ya!

Download di :<https://github.com/timedoorAcademy-smp/intl-catch-the-bird-template.git>

**Lalu buka template tersebut di visual studio code!**

Seperti biasa, tampilannya sudah dibuat dalam HTML dan CSS. Kita hanya perlu membuat logika permainannya dengan JavaScript!

## 2. Code

**Yuk kita mulai!**

### 1. Initialize variable

**Yuk tulis code ini dibawah tag <script>**

Dalam game ini kita akan menggunakan empat variable :

1. Variable showScore : Menampung elemen HTML untuk menampilkan score.
2. Variable bird : menampung elemen bird dalam HTML.
3. Variable score untuk menampung nilai score
4. Variable time untuk menampung nilai waktu.



#### Codes

```
//1. Set Variable
const showScore = document.querySelector('.scoreGame');
var bird = document.getElementById('player');
var score = 0;
var time = 0;
```



## 2. Function randomMove

Dalam game ini, burung akan bergerak secara random.



Dalam Scratch, bagaimana cara membuat sprite bergerak secara acak?

**Untuk membuat sprite bergerak secara acak, kita harus mengatur posisi (koordinat x dan y) menjadi random.**

Tulis code ini setelah code inisialisasi variable!



### Codes

```
//2.move bird in a random position
function randomMove(){
 var x = Math.floor(Math.random()*481);
 var y = Math.floor(Math.random()*1052);
 bird.style.top = x + 'px';
 bird.style.left = y + 'px';
}
```



Apakah kamu paham logika function diatas?

Dalam function diatas, nilai x diatur menjadi nilai acak dari 0-181.

Nilai y diatur menjadi acak dari 0-1052.

Kemudian posisi x burung diacak menjadi variable x.

Dan posisi y diacak menjadi variable y.



**Simpan code-mu lalu jalankan program!**

Dapatkan burung berpindah tempat secara acak?



### 3. Loop Random Move



**Menurutmu, kenapa burungnya belum dapat berpindah?**

Masih ingat bagaimana cara membuat sprite bergerak secara random terus-menerus?

Tulis code ini setelah function `randomMove()`



#### Codes

```
// 3. Loop random move
var transition = setInterval(function(){
 randomMove();
 time++;
}, 1000);
```



Simpan dan jalankan program.

Apakah sekarang burungnya bisa berpindah tempat?

method `setInterval` berfungsi untuk memanggil function berulang kali dalam jeda tertentu, misalnya 1000 ms.

Jadi setiap 1 detik, function `randomMove` dipanggil dan variable `time` bertambah 1.

### 4. Click, Event Listener



Ketika burung dikklik, apakah scorenya bertambah?

Masih ingat tentang event?

Tulis code ini setelah loop random move.



#### Codes

```
// 4. Click event listener
bird.onclick=function(){
 score++;
 showScore.innerHTML = `${score}`;
};
```



Syntak event listenernya terlihat berbeda ya?  
Event listerner juga bisa ditulis seperti diatas.

Dalam code diatas, event dan define functionnya ditulis menjadi satu.

## 5. Win and Lose Condition



Sudah kah scorenya bertambah ketika burung diklik?  
Apakah ada kondisi menang atau kalahnya?

Sudah kah scorenya bertambah ketika burung diklik?  
Apakah ada kondisi menang atau kalahnya?

### **Yuk tambahkan kondisi menang dan kalah!**

Jika waktu sudah 10 detik,  
jika score lebih dari sama dengan 5 maka menang  
Selain itu, kalah

Tambahkan code ini dalam variable `transition`  
dibawah code `time++`



### Codes

```
if(time == 10){
 if(score >= 5){
 document.body.style.backgroundImage =
 "url(image/win.jpg)";
 }
 else{
 document.body.style.backgroundImage =
 "url(image/lose.png)";
 }
}
```



## 6. Reset Game



Coba mainkan gamenya sampai win!  
Apakah burungnya masih berpindah ?



Masih ingat kode **stop all** di scratch?  
Tambahkan kode dibawah ini untuk me-reset  
permainan ketika kalah atau menang.



### Codes

```
function reset(){

 //perintah untuk menghentikan perulangan

 clearInterval(transition);
 bird.style.display='none';
 document.getElementById('scoreMain')
 .style.display='none';

}
```



Simpan dan jalankan program!  
Apakah permainan sudah berhasil reset ketika kalah  
atau menang?

**Jangan khawatir jika permainan blm juga reset.  
Ingin ya, function baru di define, belum di panggil.**



### Do it Your Self !

Panggil function reset dalam win and lose condition.  
Jika score >=5, panggil function reset  
Selain itu, panggil function reset.



Hari/Tanggal : \_\_\_\_\_

1. Ada berapa banyak function yang digunakan dalam Catch the Bird?

.....  
.....  
.....

2. Apa itu JavaScript Engine?

.....  
.....  
.....

3. Menurutmu, bagaimana urutan eksekusi program dalam Catch the Bird? Function mana yang dijalankan terlebih dahulu?

.....  
.....  
.....

4. Apa fungsi method `setInterval`?

.....  
.....  
.....



**Timedoctor**  
Coding Academy



# 8

## SUMMARY AND TEST



### Apa Yang Akan Kita Pelajari?

1. Review materi meeting 1 sampai 7
2. Intermediate 1 Middle Test





# ARTIFICIAL INTELLIGENCE WITH MATCH GAME



## Apa Yang Akan Kita Pelajari?

1. Pengenalan Artificial Intelligence dan cara kerjanya
2. Variable dalam JavaScript
3. Project Match Game menggunakan variable JavaScript



# ARTIFICIAL INTELLIGENCE WITH ULTRASONIC SENSOR



## Warm Up!



Pernahkah kamu menggunakan Siri atau Google Assistant? Tahukah kamu, Siri dan Google Assistant merupakan contoh dari **Artificial Intelligence**?



Hi, how can I help?



## Vocabulary

### Artificial Intelligence (AI)

Artificial Intelligence adalah kecerdasan buatan yang diprogram untuk membantu pekerjaan manusia.



Selain Siri dan Google Assistant, apakah kamu tahu contoh AI lainnya?



## Check This Video



<http://bit.ly/meeting9-AI>



Penerapan AI lainnya yang sering kita temui adalah **saran pencarian pada Google**. Saat kita mengetik sesuatu di dalam kolom search Google, maka Google akan memunculkan rekomendasi dari hal yang sedang kita cari. Contoh lainnya adalah face recognition.



Semakin lama, **komputer semakin pintar**. Mungkin ti akan muncul self driving car atau alat pendekripsi penyakit yang **lebih baik daripada manusia**.



### Kenapa hal ini mungkin terjadi?

Karena saat ini, satu dari jenis Artificial Intelligence sudah berkembang yaitu **Machine Learning**.



## Machine Learning



### Vocabulary

#### Machine Learning

Mesin yang dikembangkan untuk bisa belajar dengan sendirinya tanpa arahan dari penggunanya.



Bagaimana cara nya ya agar komputer bisa belajar dengan sendiri?

**Setelah ini kita akan mencoba mengembangkan model machine learning dan mencari tahu bagaimana cara kerjanya!**

# Pertemuan 9- Artificial Intelligence with Match Game

## MACHINE LEARNING MODEL



### How does Machine Learning Works?



Input Data



Analyze Data



Finding Pattern



Prediction

**Learning come from experience**, merupakan konsep pembelajaran machine learning.

Machine learning mengambil banyak data, kemudian mempelajari data tersebut untuk mengenali pola. Lalu dari pola tersebut digunakan untuk membuat prediksi dari sebuah data.



### AI Face Mask Prediction



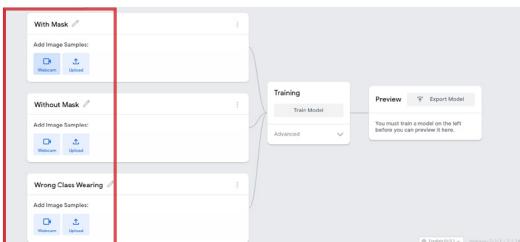
Setelah tahu cara kerja Machine Learning, **yuk kita buat model machine learning yang bisa mendeteksi orang memakai masker atau**

Kita akan menggunakan tools yang dikembangkan oleh Google!

Go to : <https://teachablemachine.withgoogle.com/train/image>

#### Step 1: Create Label

Kita akan membuat **3 jenis label** yang akan di prediksi oleh mesin, yaitu: **With Mask**, **Without Mask** dan **Wrong Mask Wearing**

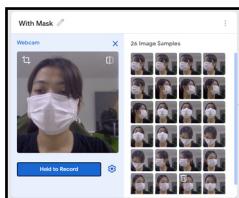




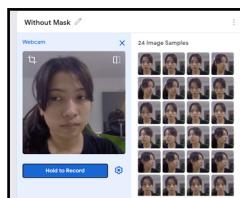
## Step 2 : Input Data

Sekarang saatnya **memasukkan data ke tiap label yang telah dibuat**. Data yang akan dipakai adalah gambar diri kita sendiri yang diambil melalui **webcam**.

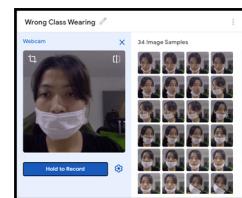
**Klik webcam** lalu klik **Hold to record** untuk mengambil gambar.



With mask



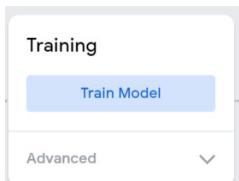
Without Mask



Wrong Wearing Mask

## Step 3: Training Model

Setelah memberikan data ke masing-masing label, sekarang saatnya model untuk **menganalisis dan menemukan pola dari data** yang disebut dengan **Training Model**



Klik **Train Model** dan tunggu hingga proses selesai !

## Step 4: Prediction

Sekarang model yang kita buat sudah **siap memprediksi** data

Pada **Preview, aktifkan input** dan lihat apakah model yang kita buat dapat **memberikan prediksi yang tepat?** baru.



## Variable in JavaScript



Masih ingatkah kamu **apa itu Variabel?**

Jika diilustrasikan variabel akan terlihat seperti ini :



Empty variable



Variable with melon juice



Variable with hot chocolate

Gelas pertama adalah **variable yang tidak diberikan nilai** (dalam javascript bisa disebut **undefined**).

Pada gelas berikutnya berisikan dua jenis minuman yang berbeda dan sama halnya dengan **variable yang dapat menampung nilai apapun** baik tipe data berupa number, string, maupun object.



Masih ingat cara menggunakan **variable** di Js?



## Codes

```
var x;
var y="Andrea";
var z=100;
```



Berapa nilai variabel x?  
Apa nilai variabel y?  
Berapa nilai variabel z?



Masih ingatkah kamu **deklarasi variabel**?



Sebelum tahun 2015, hanya keyword **var** yang bisa digunakan untuk mendeklarasikan variable. Lalu, setelah tahun 2015 untuk mendeklarasikan variabel bisa menggunakan keyword **var**, **let**, dan **const**.



## Var Keyword

Dalam project-project sebelumnya , kita sudah pernah menggunakan keyword `var` untuk mendeklarasikan variable.



### Codes

```
var myName = "Andrea";
var myAge = 15;
```

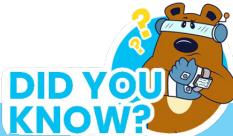


### Codes

```
var myName = "Ayu";
var myName = "Wanda";
console.log(myName);
```

```
-
2 var myName = "Ayu";
3 var myName = "Wanda";
4 console.log(myName);
```

RESULT VIEW X CONSOLE X  
Wanda



Tapi, tahukah kamu kenapa keyword `let` dan `const` diciptakan?



Kira-kira, apa yang akan tampil dalam console?

Dalam coding disamping, variable `myName` menghasilkan output "Wanda". Meskipun deklarasi awal variable nama diberikan nilai "Ayu". Ini disebut sebagai **reassign** karena memiliki identifiers (nama variable) yang sama.



DID YOU  
KNOW?

Reassign adalah cara penulisan kode yang buruk. Karena jika dalam coding yang kompleks maka proses debugging ketika programnya error akan menjadi sulit.



### Let Keyword

Let adalah fitur baru dari pengembangan dari var.

```
1 let myName = "Ayu";
2 let myName = "Wanda";
3 console.log(myName);

RESULT VIEW × CONSOLE ×
error: unknown: Identifier 'myName' has already been declared (2:4)
```

Jika menggunakan keyword let untuk deklarasi lebih dari 1 variable dengan identifiers yang sama, maka akan terjadi error

karena variable myName telah dideklarasikan sebelumnya sehingga kita tidak dapat melakukan deklarasi variable dengan identifiers yang sama,



### Const Keyword

Sesuai dengan singkatannya yakni konstanta yang berarti tetap. Const digunakan untuk variabel yg nilainya statis(tetap) atau tidak berubah.

**Const = Konstanta = Tetap**



Masih ingat kapan kita perlu menggunakan keyword const? Yuk kita ingat-ingat lagi penggunaan keyword const.

Const harus diberi nilai ketika dideklarasikan. Pemberian nilai ke variabel disebut inisialisasi. Nilainya tidak hanya berupa angka atau string, tapi bisa juga berupa object, misalnya arduino atau led.

### Syntax!

```
const minutesPerHour = 60; // correct

// incorrect
const minutesPerHour;
minutesPerHour = 60;
```



## Quiz

1. Masih ingat tidak bagaimana aturan penulisan identifiers dalam javascript ? Coba sebutkan ya!

.....  
.....  
.....

2. Dibawah ini, manakah penulisan identifiers yang benar?

1MyName

myName

My Name

my\_name

## Project 1

### Match Game!



Pernah tidak kamu memainkan match game?  
Dimana kamu harus menyocokkan kartu dengan gambar yang sama?

Cek gamenya dibawah ini ya!

<https://timedoors-match-game.netlify.app/>

Nah , kali ini kita akan membuat game itu menggunakan konsep variable!

# Pertemuan 9- Artificial Intelligence with Match Game

## MATCH GAME

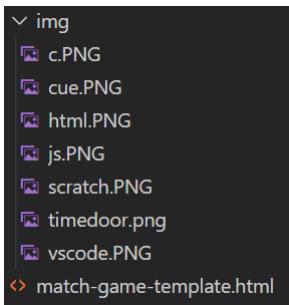


Yuk mulai kerjakan!

### 1. Prepare Template

Download template di <https://github.com/timedoorAcademy-smp/int1-9-match-game-template.git>

Lalu buka folder tersebut di visual studio code!



Dalam templat tersebut sudah terdapat **folder img** yang berisi gambar-gambar yang digunakan dalam kartu. **File match-game.html** adalah tempat mu untuk menulis code javascriptnya!



Coba buka file match-game.html, lalu jalankan programnya. Apakah tampilannya sudah ada? Coba klik kartunya, apa yang terjadi?



Tidak ada yang terjadi, karena file tersebut baru berisi HTML+CSS, saatnya kamu memberikan code JavaScriptnya!

### 2. Code

#### **Yuk kita mulai!**

Semua code diketik dalam tag `<script>...</script>`



## 1. Initialize variable

Dalam program ini, diperlukan beberapa variable.

### 1. Variable cards

menyimpan semua gambar dalam elemen HTML pada class memory-card.

### 2. Variable hasFlippedCard

Variable ini bertipe boolean (true/false). jika kartu sudah terbuka nilainya akan diset menjadi true, jika kartu belum terbuka maka nilainya false.



`hasFlippedCard = False`



`hasFlippedCard = True`

### 3. Variable lockBoard.

Variable ini juga bertipe boolean. Jika 2 kartu yang terbuka memiliki gambar yang sama maka nilai lockBoard akan diset true(kartu dikunci). Jika tidak, maka nilainya diset false.



`lockBoard = True`



`lockBoard = False`

### 4. Variable firstCard : menampung kartu pertama

### 5. Variable secondCard : menampung kartu kedua

Yuk tulis code ini dibawah tag `<script>`



### Codes

```
//1. Set Variable
const cards = document.querySelectorAll
('.memory-card');
var hasFlippedCard = false;
var lockBoard = false;
var firstCard, secondCard;
```

## 2. Click Event Listener



Masih ingat kan tentang **event listener**?

Kira-kira tipe event apa yang digunakan dalam game ini?

Dalam game ini, kita akan memberikan event click ke semua elemen kartu. Karena ada banyak elemen kartu, maka semua kartu akan dianggap sebagai object kartu.



**Apa itu object kartu?**

Tidak usah bingung ya! Kita akan mempelajari lebih banyak tentang objek di pertemuan selanjutnya!

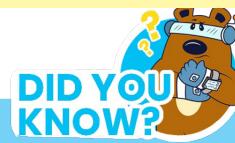
Tulis code ini dibawah inisialisasi variable.



### Codes

```
//2. Event Click
cards.forEach(card =>
 card.addEventListener('click', flipCard)
);
```

Code diatas berfungsi untuk membuat agar seluruh kartu dapat di klik dan setelah di klik akan menjalankan function flipCard.



`forEach` merupakan metode untuk melakukan perulangan dari jumlah elemen array.



**Simpan dan jalankan program.**

Apa yang terjadi ketika kartu di click?



### 3. Function flipCard

**Tidak ada yang terjadi ketika kartu diklik**, karena function yang dipanggil belum di define. Yuk coba define function flipCard.

Tulis code ini dibawah event listener



#### Codes

```
function flipCard() {
 // first condition
 if (lockBoard)
 return;
 // second condition
 if (this === firstCard)
 return;
 this.classList.add('flip');
 // third condition
 if (!hasFlippedCard) {
 hasFlippedCard = true;
 firstCard = this;
 return;
 }
 secondCard = this;
 checkForMatch();
}
```



#### Bagaimana logika function diatas?

Bisakah kamu membayangkan apa yang dijalankan dari function diatas? Ingat-ingat lagi tentang konsep return.

Function diatas menjalankan 3 kondisi :

1. Kondisi pertama berfungsi untuk membuat player tidak bisa klik kartu yang sudah terkunci
2. Kondisi kedua berfungsi untuk membuat player tidak bisa klik kartu pertama yang sudah terbuka.
3. Kondisi ketiga, berfungsi untuk mengecek apakah tidak ada kartu yang terbuka, jika kondisi benar artinya ada satu kartu yang terbuka lalu kartu tersebut diset menjadi first card.

Selain semua kondisi itu, kartu kedua akan yang diklik menjadi secondCard dan dicek kecocokan 2 kartu tersebut.



**Simpan dan jalankan program.**  
Dapatkan kartu terbuka ketika diklik?

#### 4. Function checkForMatch

Dalam function flipCard terdapat **function checkForMatch** yang belum di define. Function ini berfungsi untuk mengecek apakah 2 kartu yang terbuka memiliki gambar yang sama.

Tulis code ini dibawah ini!



#### Codes

```
function checkForMatch() {
 // 1. set variable
 let isMatch = firstCard.dataset.framework
 === secondCard.dataset.framework;

 //2. check condition
 if(isMatch == true){
 disableCards();
 }
 else{
 unflipCards();
 }
}
```



**Bagaimana logika dari function diatas?**

1. Variable isMatch berisi nilai firstCard sama dengan secondCard.
2. Jika isMatch bernilai benar, maka kartu akan dikunci.  
Selain itu, kartu akan ditutup kembali



isMatch= True



isMatch= False



## 5. Function disableCards

Dalam function checkForMatch terdapat pemanggilan **function disableCard** yang belum di define. Function ini berfungsi untuk menghilangkan event listener dari kartu dengan kondisi 2 kartu terbuka dengan gambar yang sama .

Tulis code ini dibawah ini!



### Codes

```
//remove event listeners
function disableCards() {
 firstCard.removeEventListener('click', flipCard);
 secondCard.removeEventListener('click', flipCard);
 resetBoard();
}
```



### Simpan dan jalankan program.

Coba klik 2 kartu yang berbeda, apakah kartu kembali tertutup?

## 5. Function disableCards

Dalam function checkForMatch terdapat pemanggilan **function unFlipCards** yang belum di define. Function ini berfungsi untuk menutup kembali 2 kartu terbuka yang gambarnya tidak sama.

Tulis code ini dibawah ini!



### Codes

```
function unflipCards() {
 lockBoard = true;
 setTimeout(() => {
 firstCard.classList.remove('flip');
 secondCard.classList.remove('flip');
 resetBoard();
 }, 1500);
}
```



`setTimeout` merupakan metode untuk memanggil function setelah beberapa milidetik.



### Simpan dan jalankan program.

Coba klik 2 kartu. Setelah itu, apakah kamu bisa klik kartu yang lainnya?

## 6. Function resetBoard

Jika setelah klik2 kartu kamu tidak bisa klik kartu lainnya setelah jangan panik! Itu terjadi karena nilai variable belum di reset.



Apakah kamu melihat function resetBoard di function sebelumnya?

**Function resetBoard** belum di define. Function tersebut digunakan untuk mengatur ulang nilai variable ke nilai awalnya.



Do it Your Self !

Dalam function resetBoard, set nilai hasFlippedCard dan lockBoard menjadi false firstCard dan secondCard menjadi null



Codes

```
function resetBoard() {
 // set hasFlippedCard
 // set lockBoard
 // firstCard
 // secondCard
}
```



### Simpan dan jalankan program.

Apakah kamu sudah bisa membuka semua kartu?  
Apa yang kurang dalam game ini?

## 6. Shuffle Cards

Tidak seru jika posisi kartunya tidak acak! Yuk kita acak posisi kartunya!



### Codes

```
(function shuffle() {
 cards.forEach(card => {
 let randomPos = Math.floor(Math.random() * 12);
 card.style.order = randomPos;
 });
}());
```

Variable randomPos berisi nilai acak dari 0-12.  
Kemudian susunan kartu diacak menjadi randomPos



### Simpan dan jalankan program.

Apakah susunan kartu sudah acak?  
Jika iya, maka **program selesai!**

## Pertemuan 9- Artificial Intelligence with Match Game

### CHECK YOUR UNDERSTANDING



Hari/Tanggal : \_\_\_\_\_

1. Bagaimana prinsip kerja machine learning?

.....  
.....  
.....

2. Menurutmu, bagaimana machine learning bisa mengatasi masalah lingkungan?

.....  
.....  
.....

- 3.
- ```
const cards = document.querySelectorAll
('.memory-card');
var hasFlippedCard = false;
```

Menurutmu, mengapa inisialisasi variable `cards` menggunakan keyword `const`, sedangkan keyword `hasFlippedCard` menggunakan keyword `var`?

.....
.....
.....



4.

```
if (!hasFlippedCard) {  
    hasFlippedCard = true;  
    firstCard = this;  
    return;  
}
```

Mengapa ketika kondisi `!hasFlippedCard` benar, variable `hasFlippedCard` di set menjadi true?

.....

.....

.....



10

MEDIA, YOUR BRAIN AND EXPRESSION



Apa Yang Akan Kita Pelajari?

1. Mengetahui penyebab kecanduan sosial media
2. Mengetahui penggunaan expression (operator) dalam JavaScript
3. Membuat web calculator dengan JavaScript



MEDIA, YOUR BRAIN AND EXPRESSION



Warm Up!



Apa **caption yang tepat** untuk gambar disamping?



Sosial media apa yang paling sering kamu buka di smartphone mu?

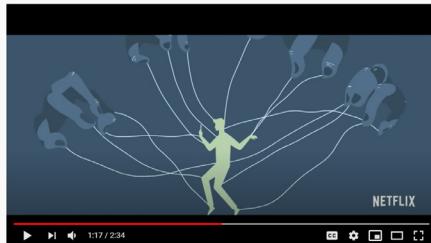
Menurutmu, apa yang membuat kamu **betah lama** menggunakan social media itu?



Sering kali, kamu tidak punya alasan atau tujuan untuk menggunakan smartphone. Kamu melakukannya karena bermain smartphone sudah menjadi **habit, fitur-fitur di sosial media** memang **dirancang agar menarik perhatianmu** dan kamu betah berlama-lama menggunakannya!



Check This Video



The Social Dilemma | Official Trailer | Netflix

Otak manusia memproduksi **hormon dopamine** yang dihasilkan ketika kita **menerima informasi menyenangkan**, salah satunya ketika kita menerima likes atau comment pada post di sosial media. Rasa senang yang timbul saat bermain sosial media itu muncul karena sosial media memiliki **addictive design**.



Dopamine

Senyawa kimia yang menyebabkan rasa senang.

Addictive Design

Design aplikasi/sosial media yang membuat orang ketagihan, seperti likes, rekomendasi, dll yang mengaktifkan dopamine.



Apakah media sosial yang kamu suka memiliki addictive design?



Social Media Check

Sekarang coba check apakah 3 sosial media tersebut mempunyai fitur addictive design!

Beri tanda check (✓) jika ada

Fitur Addictive Design	1.	2.	3.
Rewards (Like/Comment)			
Bisa share ke teman			
Auto Play			
Konten/feeds yang banyak			
Auto refresh feeds			
Rekomendasi feeds yang sejenis			

"Only two things who called the customers as users; drug and social media"



Bagaimana **strategi kamu** agar tidak **terlarut dalam addictive design** dari sosial media?

.....
.....
.....



What Are Expressions?



Masih ingat tidak **apa itu operator**?
Ada berapa **jenis operator** dalam coding?



Ingat juga **apa itu variable**?

Kali ini kita akan membuat **project kalkulator** yang menggunakan **expressions dalam JavaScript**



Vocabulary

Expressions

Gabungan dari variable dan operator yang menghasilkan suatu nilai.



Contoh expression sudah sering kita jumpai loh!
Contohnya etika kita memberi nilai ke variable.
Misalnya, `x = 10` yaitu memberikan nilai 10 ke variabel x.

Kalian bisa menganggap expression sebagai sebuah formula

Contoh expression lainnya adalah seperti code disamping!



Example

```
velocity = speed/time;
```

KIND OF EXPRESSIONS



Kind of Expressions

Sama seperti operator dan variable, **expressions dibagi jadi 3 jenis** :



1. Arithmetic Expressions



Masih ingat apa itu **operator arithmetic**?

Arithmetic Expression adalah expression yang menggunakan angka dan **operator aritmatika** (+, -, *, /, dll) untuk menghasilkan suatu nilai berupa angka.



Example

```
total = x*x + 2*x + 1;
```



2. Logical Expressions



Coba sebutkan apa saja yang termasuk **operator logika**?

Logical Expression adalah expression yang menggunakan **operator logika** (&&, ||, !, dll) untuk menghasilkan **nilai benar atau salah**.



Example

```
die = touchingBullet || collision;
```



Ada yang bisa membaca expression diatas?



3. String Expressions



Masih ingat apa itu **string**?

String Expression adalah expression yang menggunakan operator , yang biasanya digunakan untuk menggabungkan string dengan string lain atau dengan nilai dari sebuah variable.



Example

```
score = 70;  
say = "Your score is", +score;
```



Kira-kira dari code diatas, apa nilai dari variable say?



Numbers In JavaScript



Sebenarnya **JavaScript tidak mengenal tipe data** (misal integer, float, dll), melainkan hanya mengenal **number/angka**.

Example

```
// integer
var score = 3;

// decimal, float
var amount = 39.99;
```

Dalam **variable javascript** dapat diberi nilai berupa **bilangan bulat** ataupun **bilangan desimal**.



Datatype Conversion Functions

Seperti yang kita tahu, JavaScript hanya mengenali **tipe data string** dan **angka** (bilangan bulat dan desimal). Dalam JavaScript terdapat **function yang bisa digunakan untuk mengubah tipe data** lho!

Perhatikan function dibawah ini!

Function	Description
<code>parseInt()</code>	Mengubah string menjadi integer
<code>parseFloat()</code>	Mengubah string menjadi desimal
<code>String()</code>	Mengubah integer menjadi string



Operator Precedence



$3 * 10 - 15 : (3 + 2) = \dots$

Berapa hasil dari soal diatas?

Dalam soal matematika diatas, **yang mana harus dikerjakan terlebih dahulu**? perkalian? pembagian? atau pengurangan?

Dalam matematika ada **prioritas operator** yang menentukan operator mana yang harus dikarjakan terlebih dahulu, **begitu juga dalam JavaScript!**

Perhatikan tabel dibawah ini ya!

Tabel ini berisi urutan prioritas operator dalam JavaScript (highest precedence to lowest)

Type	Operator
parenthesis	()
negation/increment	!, ++, --
multiply/divide	*, /, %
addition/subtraction	+, -
comparison	<, <=, >, >=
equality	==, !=
logical-and	&&
logical-or	
assignment	=, +=, -=, *=, /=, %=



JavaScript External File

```
<html>
  <body>
    <script>
      console.log('Hello world!');
    </script>
  </body>
</html>
```

Biasanya kita menulis coding JavaScript dalam file HTML. Tapi, kita bisa juga lho **menulis coding JavaScript secara terpisah dari file HTML**.

Cara ini biasanya digunakan **pada project besar** karena dengan cara ini dapat lebih mudah mengelola coding project.



Scr attribute

```
belajar-js/
└── kode-program.js
└── index.html
```

Untuk menggunakan JavaScript eksternal file, kita harus membuat 2 buah file yaitu file **HTML** dan **JavaScript** dalam **satu folder**.

Karena kedua file tersebut terpisah, jadi kita harus **menghubungkan 2 file itu**.

Untuk itu, kita perlu menggunakan **atribut src** yang dituliskan di dalam tag `<script...></script>`

Syntax!

```
<script src = "fileName.js" ></script>
```

Dari sintaks diatas, kamu hanya perlu mengubah “`fileName`”.js dengan nama file JavaScript yang kita buat.



HTML File

```
calculator.html
<html>
<head>
    <title>Kalkulator</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="main-container"> ...
</body>
</html>
<script src = "calculator.js">
</script>
</body>
</html>
```

```
calculator.js
function calculator() {
    //variable angka satu
    var firstNumber = parseInt(document.getElementById("number1"));
    //variable angka dua
    var secondNumber = parseInt(document.getElementById("number2"));
    //variable operasi aritmatik
    var op = document.getElementById("op").value;
    //variable angka hasil
    var result = document.getElementById("result");
    //variable untuk total angka satu dan angka dua
    var total;
    if (isNaN(firstNumber) || isNaN(secondNumber)) {
        alert("Fill the form with number!");
    } else {
        if (op == "+") {
            total = firstNumber + secondNumber;
        } else if (op == "-") {
            total = firstNumber - secondNumber;
        } else if (op == "*") {
            total = firstNumber * secondNumber;
        } else if (op == "/") {
            total = firstNumber / secondNumber;
        }
    }
}
```

Js File

Coding Practice!

Main Project

My Calculator

Mau bikin kalkulator sendiri tidak?

Yuk buat kalkulator mu sendiri menggunakan **kONSEP EXPRESSION DALAM JAVASCRIPT**!

Cek projectnya di <https://timedoors-mycalculator.netlify.app/>

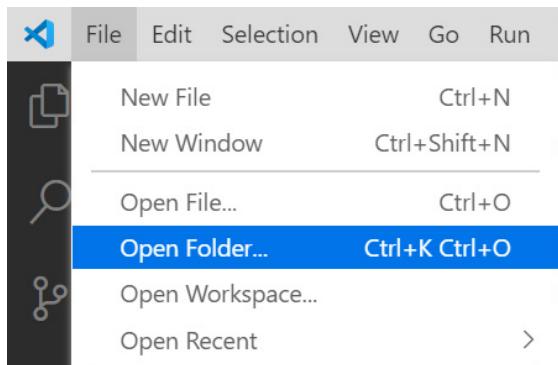
Yuk mulai kerjakan!

1. Prepare Template

1. Buka link di browser <https://github.com/timedoorsAcademy-smp/meeting10-calculator-template>
2. Extract file yang di download kemudian buka folder template di **Visual Studio Code**

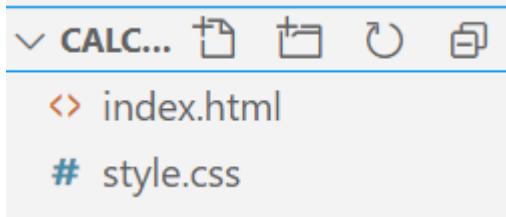
Pertemuan 10- Media, Your Brain and Expression

MY CALCULATOR

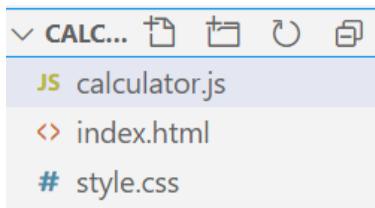


Klik **File** lalu pilih **Open Folder** kemudian **pilih folder template** yang telah di extract.

Terdapat **2 buah file** di dalamnya, yaitu file HTML dan file CSS.
Sehingga kita perlu membuat 1 file baru untuk file JavaScript!



3. Klik **File** lalu pilih **New File** untuk membuat file baru.
 4. **Save file** baru tersebut dengan nama **calculator** dan **save as type JavaScript**. Pastikan kamu menyimpannya di folder yang sama.
- Jika berhasil, akan ada **3 file baru dalam folder mu!**





2. Coding

Sekarang, dalam file **calculator.js** tuliskan code berikut ini.



Code

```
function calculator() {  
  
    var firstNumber = parseInt(document.  
getElementsById("number1").value);  
    var secondNumber = parseInt(document.  
getElementsById("number2").value);  
  
    var operator = document.getElementsById("op").value;  
    var result = document.getElementsById("result");  
  
    var total;  
  
    if (isNaN(firstNumber) || isNaN(secondNumber)) {  
        alert('Fill the form with number!')  
    } else {  
        if (operator == "+") {  
            total = firstNumber + secondNumber;  
        } else if (operator == "-") {  
            total = firstNumber - secondNumber;  
        } else if (operator == "x") {  
            total = firstNumber * secondNumber;  
        } else if (operator == "/") {  
            total = firstNumber / secondNumber;  
        }  
    }  
    result.value = total;  
}
```



Coba cek coding calculator tadi! Disana kita menggunakan **function isNaN()** .isNaN() merupakan sebuah function yang berfungsi untuk mengecek apakah nilai dari expression itu bukan angka.

Setelah kita memiliki file Js, **sambungkan file HTML dengan JavaScript**.

Buka file index.html lalu lengkapi tag <script>... /<script>



Codes

```
<script src = "calculator.js" ></script>
```

3. Save

Simpan kedua file dengan klik file lalu pilih Save.

4. Run Code

Sekarang coba pilih file HTML nya lalu **jalankan program di browsermu** dengan menekan **Alt + B** dan coba test apakah programmu sudah berjalan dengan benar?



Hari/Tanggal : _____

1.

Apa yang menyebabkan kita betah berlama-lama bermain sosial media?

.....
.....
.....

2.

Apa itu expression dalam coding? Coba berikan satu contoh expression yang digunakan dalam project my Calculator.

.....
.....
.....

3.

Perhatikan expression dibawah ini! Mengikuti aturan operator precedence, berapa nilai dari variable a?

a= 60/4*(8-2)+10;

.....
.....
.....

4.

Kapan kita perlu menggunakan javascript external file?

.....
.....
.....



11

DIGITAL TRACKING AND ARRAY



Apa Yang Akan Kita Pelajari?

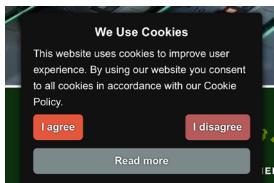
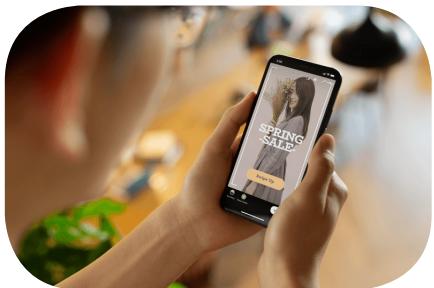
1. Mengetahui digital tracking
2. Mengetahui penggunaan array dalam javascript
3. Membuat web generator dengan JavaScript



DIGITAL TRACKING AND ARRAY



Warm Up!



Informasi apa yang sering kamu share di internet/sosial media?

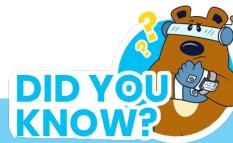


Pernah tidak kamu mendapat iklan berdasarkan apa yang baru saja kamu cari di internet??

Kenapa ya hal itu bisa terjadi?
internet??



Pernah tidak kamu diminta untuk mengaktifkan cookies saat membuka sebuah website?



Setiap kali kamu menggunakan internet, semua aktivitas yang kamu lakukan di internet akan ditrack dan disimpan.



Vocabulary

Online Tracking

Aplikasi/website yang mengumpulkan informasi aktivitas online .



Check This Video



Selain aktivitas **onlinemu**, app/ website juga akan menyimpan lokasimu(GPS), dan data profilmu (nama, email, no hp, dll). Terkadang kamu juga tanpa sadar **mengaktifkan Cookies** yang ada di web browsermu.



Kenapa ya hal itu bisa terjadi?



Vocabulary

Cookies

File teks kecil yang disimpan di komputer yang melacak apa yang dilakukan seseorang di situs web

Situs web atau aplikasi akan mengirimkan kamu iklan berdasarkan informasi yang mereka kumpulkan tentang dirimu.

"if the seller does not have anything to sell, then you're their product"



Lalu apa yang harus kita lakukan?

If You Want to Limit What Is Tracked.....



Turn Off Cookies



Adjust Privacy Setting



Limit What You Share

Pertemuan 11- Digital Tracking and Array

ARRAY



Array



Warm Up!

Bayangkan kamu diminta untuk membuat aplikasi web dan ingin menampilkan nama-nama buah.



Codes

```
var fruits1 = "apple";
var fruits2 = "pinapple";
var fruits3 = "mango";

document.write(fruits);
document.write(fruits);
document.write(fruits);
```



Apakah bisa dengan coding seperti disamping?

Bagaimana kalau ada 100 buah, apakah kita akan membut variabel sebanyak 100? **Lelah dong?**

Karena itu kita bisa menggunakan **array(list)**.



Masih ingat apa itu array?

Array merupakan **wadah** digunakan untuk menyimpan **sekumpulan data** dalam satu tempat.

Setiap data dalam Array **memiliki indeks**, sehingga kita akan mudah memprosesnya. **Indeks array** selalu dimulai dari **nol (0)**

apple	pinapple	mango
index ke 0	index ke 1	index ke 2



Hari ini, kita akan membuat web yang bisa meng-generate quotes secara random menggunakan array!



Creating an Array



Masih ingat tidak cara membuat array?

Syntax!

```
var arrayName = [ ];
```

Untuk membuat array, di awali dengan keyword var di ikuti dengan nama array dan menggunakan kurung siku [...]



Codes

```
var fruits = [ ];
```



Apa **nama array** dalam coding disamping?
Apa **nilai dari array** tersebut?



Array fruits tersebut **masih kosong**.

Lalu bagaimana cara mengisi elemen ke array tersebut?

Syntax!

```
var arrayName = [ item1, item2, item3, ... ];
```

Kita bisa mengisi data ke dalam array yang ditulis **di dalam tanda kurung siku [...]** lalu setiap data **dipisah dengan tanda koma (,)**.



Let's Practice!



Buka <https://playcode.io/> dibrowsermu! Lalu pada **script.js** tuliskan code untuk **membuat array bernama fruits** yang **berisi 5 elemen** (apple, banana, dragonfruit, pineapple, mango)

Kalau sudah selesai tunjukkan ke gurumu ya!



Access the Element of an Array



Dari array yang telah dibuat, bagaimana cara kita mengambil elemen "Pineapple"?

Syntax!

```
arrayName[index];
```



Example

```
document.write(fruits[1]);
```



Apa elemen yang akan ditampilkan dari coding disamping?



Let's Practice!

Menggunakan playcode.io. Coba **tampilkan index ke 2** dalam array yang kamu buat tadi!



Insert Element to an Array



Sebelumnya kita sudah **membuat array dengan 5 elemen** didalamnya. Tahukah kamu **bagaimana caranya menambah elemen baru** ke array tersebut?



Ada dua cara untuk menambah data ke dalam array :

- Menggunakan indeks;
- Menggunakan metode `push()`.



1. Menggunakan Index

Misalnya, kita punya array seperti dibawah ini!



Example

```
var fruits = ["apple",  
"banana", "dragonfruit", "pineapple", "mango"];
```



Ada berapa elemen di array tersebut?

Kita ingin menambahkan **elemen watermelon** pada **indeks ke-5**, maka kita bisa menggunakan cara ini :



Example

```
var fruits[5] = "watermelon";
```

Pertemuan 11- Digital Tracking and Array

INSERTING ELEMENT TO AN ARRAY



Tapi cara ini memiliki kekurangan lho!

Kita **harus tahu jumlah data atau panjang array-nya**, barulah kita bisa menambahkan elemen baru.

Apabila kita memasukan **nomer indeks sembarangan**, maka nanti yang akan terjadi adalah data yang ada di dalam indeks tersebut akan **ditindih** atau bisa juga membuat **indeks array menjadi**

Misalnya:



Example

```
var fruits = [ "apple",
"banana", "dragonfruit", "pinapple", "mango"];
var fruits[10] = "watermelon";
```

Hasilnya:

```
apple,banana,dragonfruit,pinapple,mango,,,,,,grape
```



Lalu bagaimana solusinya?



2. Menggunakan Metode Push()

Metode **push()** akan **menambahkan data** ke dalam array **ke elemen paling belakang**. Sehingga kita tidak perlu tahu berapa panjang arraynya

Syntax!

```
arrayName.push("elemen");
//add multiple element at once
arrayName.push("elemen1", "elemen2" .. dst);
```



Example

```
fruits.push("watermelon");
```



Let's Practice!

Yuk kita latihan! Masih menggunakan text editor playcode.io. **Tambahkan satu elemen baru** pada array fruits lalu tampilkan elemen tersebut! Kalau sudah selesai tunjukkan ke gurumu ya!



Deleting Array Element



Kita sudah belajar cara menambahkan elemen ke array, gimana ya kalau kita ingin **menghapus elemen dari sebuah array**?

Ada dua cara untuk menghapus data ke dalam array :

- Menggunakan delete;
- Menggunakan metode `pop()`.



1. Menggunakan Delete

Kita dapat menghapus data dengan nomer indeks tertentu dengan **delete**.

Syntax!

```
delete arrayName[index];
```

Pertemuan 11- Digital Tracking and Array DELETING ARRAY ELEMENT



Kekurangan dari delete yaitu akan menciptakan ruang kosong di dalam array.

Misalnya:



Example

```
var fruits = ["apple",
"banana", "dragonfruit", "pinapple", "mango"];
delete fruits[2]
```

Hasilnya:

apple,banana,,pinapple,mango



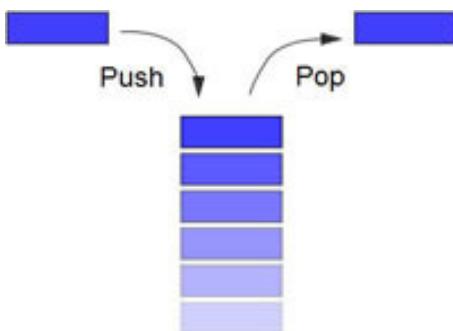
Ketemu tidak permasalahan dari hasil disamping?



2. Menggunakan Metode Pop()

Metode `pop()` merupakan kebalikan dari metode `push()`.

Metode `pop()` akan **menghapus array yang ada di paling belakang**.



Syntax!

```
arrayName.pop();
```



Example

```
var fruits = ["apple", "banana", "mango"];
fruits.pop();
```



Dari contoh coding diatas, elemen mana yang akan terhapus?



Tapi bagaimana caranya menghapus elemen tertentu ya? atau kita ingin elemen terhapus dari depan?

Menghapus Elemen dari Depan

Kita juga dapat **menghapus elemen dari depan** dengan menggunakan metode `shift()`.

Syntax!

```
arrayName.shift();
```



Example

```
var fruits = ["apple", "banana", "mango"];
fruits.shift();
```



Elemen mana yang akan terhapus?



Menghapus Elemen Tertentu

Apabila kita ingin **menghapus elemen pada indeks tertentu**, maka dapat menggunakan metode `splice()`

Syntax!

```
arrayName.splice(<index>, <total>);
```

<index> adalah indeks dari data di dalam array yang akan dihapus;
<total> adalah jumlah data yang akan dihapus dari indeks tersebut.



Example

```
var fruits = ["apple", "banana", "mango"];
fruits.splice(1, 1);
```



Elemen mana yang akan terhapus?



Changing Array Element

Kita sudah belajar cara menambahkan dan menghapus elemen array. Lalu **bagaimana ya cara mengubah elemen array?**

Kita sudah pernah mempelajarinya di Beginner. **Masih ingat?**



Example

```
var fruits = ["apple", "banana", "mango"];
fruits[1] = "grape";
```



Apa elemen index ke 1 dari array diatas? banana atau grape?



Length of Array



How to count all these fruits?



Dalam array kita bisa menambahkan banyak data, tapi bagaimana caranya **mengetahui jumlah/panjang dari array** tersebut?

Mengetahui panjang array, bisa menggunakan property `length`.

Syntax!

```
arrayName.length;
```

`length` akan memberikan nilai panjang dari sebuah array.



Example

```
var fruits = ["apple", "banana", "mango"];  
console;
```



Let's Practice!

Yuk kita latihan! Masih menggunakan text editor playcode.io.

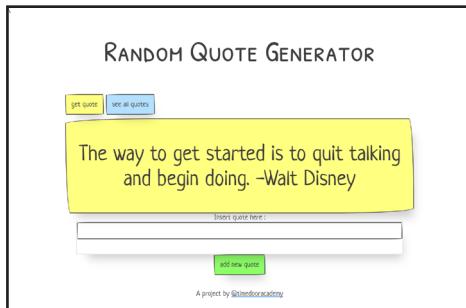
Buat array dengan 5 elemen. Lalu coba untuk

- Menghapus elemen pertama
- Menghapus elemen terakhir
- Menghapus elemen index ke 2
- Mengganti nilai emelem ke index ke 0!
- Tampilkan panjang dari array.



Coding Practice!

Random Quote Generator



Yuk cek quotes generator yang akan kita buat!

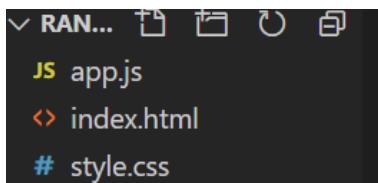
<https://quotes-timedoor.netlify.app/>

Yuk mulai kerjakan!

1. Prepare Template

1. Buka link di browser <https://github.com/timedoorAcademy-smp/meeting11-quoteGenerator-master.git>
2. Extract file yang di download kemudian **buka folder** template di **Visual Studio Code**
3. Klik **File** lalu pilih **New File** untuk membuat file baru.
4. **Save file** baru tersebut dengan nama **app.js** dengan **save as type JavaScript**. Pastikan kamu menyimpannya di folder yang sama.

Jika berhasil, akan ada **3 file baru dalam folder mu!**





2. Coding

Sekarang, dalam file **app.js** tuliskan code berikut ini.



Code

```
//keep quotes in an array
var quotes=[ "quotes 1", "quotes2", "Quotes3" ];

//function to generate random quotes from array element
function generate(){
    let random = Math.floor(Math.random() * quotes.
length);
    document.getElementById('quoteSection').
innerHTML=quotes[random];
}

//function to print all array elements
function seeAllQuotes(){
    text = "<ul>"; //create bullet
    for (i=0; i<quotes.length; i++){
        text += "<li>" + quotes[i] + "</li>";
    }
    text += "</ul>";
    document.getElementById("quoteSection").innerHTML =
text;
}
//add new element to array
function newQuotes(){
    var newQuotes = document.
getElementById("insertSection").value;
    alert("quotes added");
    quotes.push(newQuotes);
    seeAllQuotes();
}
```

Jangan lupa! Pada array quotes, ganti dengan quotes yang kamu suka atau bisa cari di internet ya! Sertakan juga sumbernya ya!



3. Save

Simpan kedua file dengan klik file lalu pilih Save.

4. Run Code

Sekarang coba pilih file HTML nya lalu **jalankan program di browsermu** dengan menekan **Alt + B** dan coba test apakah programmu sudah berjalan dengan benar?

Pertemuan 11- Digital Tracking and Array

CHECK YOUR UNDERSTANDING



Hari/Tanggal : _____

1. Apa yang menyebabkan kita mungkin mendapat iklan setelah kita searching di internet?

.....
.....
.....

2. Apa itu Cookies? Kenapa kita perlu mematikan cookies?

.....
.....
.....

3. Bagaimana array digunakan dalam web quote generator?

.....
.....
.....

- 4.
- ```
var fruits = ["apple", "banana", "mango"];
fruits.splice(1, 1);
fruits.push("pineapple");
console.log(fruits.length);
```

Apa yang akan ditampilkan di console?

Apa elemen array fruits setelah coding dijalankan?

.....  
.....  
.....



12

## THIS IS OBJECT



### Apa Yang Akan Kita Pelajari?

1. Mengetahui konsep object dalam coding
2. Mengetahui penggunaan keyword this dalam JavaScript



# OBJECT IN IOT WEB CONTROLLED



## Warm Up!



Masih ingat variable bisa diisi nilai apa saja? Kalau array bagaimana?  
Kali ini kita akan belajar tentang object

2 Dalam kehidupan nyata, bisakah kamu memberikan **contoh object?** Apa saja properties didalamnya? Aa saja yg bisa dilakukan object itu?

2 Bagaimana cara kita memodelkan object mobil dengan tipe Tesla warna putih ini di dalam coding?



Untuk menampung nilai nama mobil, model atau warna mobil sekaligus kita bisa menggunakan **object**.



### Object

Variabel tempat menampung beberapa nilai bernama (misal type, model, color) yang disebut properties dan method.



**Properties :**

type : Tesla  
model : Model 3  
color : white



**Methods**

Start()  
Stop()  
Brake()  
Drive()

Object : Car

## Properties



Dalam dunia nyata, **mobil adalah sebuah object** yang memiliki **properties** seperti nama brand, model, warna.

**Properties ini 'nilai'** yang bisa berbeda-beda. Misal ada '**Mobil**' yang **properties warna** = merah atau lainnya, tapi nama propertiesnya sama yaitu '**warna**'.

## Methods

Object '**Mobil**' juga memiliki **method** yaitu **function** misalnya '**start**', '**drive**', '**brake**', dsb. Method ini bisa digunakan di waktu yang berbeda-beda.



### DEFINE OBJECT



#### Define Object

- 
- Lalu bagaimana cara menuliskan **objek dalam JavaScript?**

Sama seperti variable dan array, object harus dibuat terlebih dahulu.

#### Syntax!

```
var objetcName = {property: "value", property2: "value", };
```

Objek pada javascript, dapat dibuat dengan tanda kurung kurawal dengan isi berupa properties dan value.



#### Example

```
var car = {
 type: "Tesla",
 model: "Model 3",
 color: "White"
};
```

Agar lebih rapi,  
penulisan object  
dapat seperti  
coding disamping!

Properties dan value dipisah dengan tanda titik dua ":"



#### Vocabulary

#### Properties

Pasangan name dan value dalam object



**Let's Practice!**

Coba tuliskan **object person** yang memiliki properties dan value seperti dibawah ini!



Name : John  
Age : 15  
Gender : Male  
Interest : Coding

Tulis code mu disini ya!



**Write here!**

Untuk mengecek apakah code yang kamu tulis diatas benar, coba tulis kembali code mu di <https://playcode.io/>. Dan tunjukkan ke gurumu!



## Pertemuan 12- This is Object ACCESSION OBJECT PROPERTIES



### Accessing Object Properties



john

Nah, tadi kita sudah membuat object person dengan beberapa properties.



Bagaimana jika kita ingin menampilkan nilai properti tertentu dari sebuah object ?

Ada **2 cara** untuk mengakses nilai dari properties :

#### Syntax!

`objectName.propertyName`

atau

`objectName[ "propertyName" ]`



#### Example

```
car.type;
car["type"]

// to print in console

console.log(car.type);
console.log(car["type"]);
```



**Let's Practice!**

Coba tuliskan cara memanggil value properties age dalam object person yang dipanggil di console.



**Write here!**

Coba tuliskan cara memanggil value properties age dalam object person yang dipanggil di console!.



**Write here!**

Untuk mengecek apakah code yang kamu tulis diatas benar, coba tulis kembali code mu di <https://playcode.io/>. Dan tunjukkan ke gurumu!



**Masih ingat tidak ?**

Selain properties, object juga memiliki method.

**Apa itu method?**

Agar lebih paham mari kita pelajari itu di meeting selanjutnya!

## Pertemuan 12- This is Object

### THIS KEYWORD



Agar lebih mengerti konsep Object, coba selesaikan coding dibawah ini.

Coba perhatikan gambar balok disamping, **bagaimana memodelkannya dengan konsep object?**



#### Code!

```
//isi titik-titik dibawah ini ya
//buat object dengan nama ruang
var ... = {
 tipe: "Balok",
 ...: "...", //deskripsikan properties warnanya
 sisi: ... //coba deskripsikan banyak sisinya
 panjang: 10,
 lebar: 5,
 tinggi: 4
};

//coba akses nilai dari panjang, lebar dan tinggi
//buat agar console menampilkan volume balok

console.log(...);

//buat agar console menampilkan volume balok
//sekarang gunakan function untuk menampilkan volumenya

function volume(){
 return "Volume balok adalah: " +;
}

console.log(volume());
```



Bagaimana, sudah mengerti bagaimana memodelkan sebuah benda dengan konsep object di coding?



## This Keyword



Jika kamu ingin menunjukkan sesuatu object/benda yang dekat, kata apa yang tepat untuk menunjukkannya?



### Vocabulary

#### This

Sebuah statement dalam program yang dapat digunakan untuk merujuk pada object tertentu dalam program bergantung pada dimana statement ini digunakan.

#### this keyword digunakan pada beberapa kondisi:

Jika berdiri sendiri, this keyword merujuk pada global object.

Jika berada didalam function, this keyword merujuk pada global object.

Jika berada didalam event, this keyword merujuk pada elemen yang menerima event tersebut.

Dalam metode call(), apply(), bind(), this keyword merujuk pada object yang akan dituju.

#### Syntax!

this

```
//this pada beberapa penggunaan
this.x = 8;
```



Agar lebih paham, kita perhatikan contoh code dibawah ini ya.



### Code!

```
var myVar = 50; //global object/variable

function coba(){
 var myVar = 500;
 console.log(this.myVar);
}

coba();
```



Apa yang akan ditampilkan pada console?

Mengapa console tidak menampilkan angka 500?

Masih ingat apa yang akan dirujuk oleh this keyword jika berada didalam function?

**Coba kita coba pahami penggunaan this keyword dengan code yang berbeda**



### Code!

```
function segitiga(){
 var alas = 6;
 var tinggi = 2;
 return alas*tinggi;
}

segitiga();
console.log(this.alas);
```



Pada code di atas, apa yang akan ditampilkan pada console?



Variable alas berada didalam function, sehingga tidak didefinisikan sebagai variable global.

Pada contoh di atas, console akan **menampilkan undefined**, hal ini dikarenakan this keyword merujuk pada global object yang belum di definisikan.



### this keyword pada Konsep Object



Bagaimana jika kamu ingin menunjukkan warna dari sebuah benda? Apa kata yang digunakan olehmu?

this keyword yang digunakan didalam konsep object, maka akan merujuk pada object tersebut atau propertiesnya.



### Code!

```
var person = {
 firstName: "Kevin",
 lastName : "Dexsyan",
 id : 100600,
 fullName : function() {
 return this.firstName + " " + this.lastName; }
};
```



Bagaimana cara kerja code di atas?

Agar lebih paham, mari kita pelajari itu di pertemuan selanjutnya!

**Untuk mengakses properties sebuah object, kita juga bisa menggunakan keyword call() dan apply()**



Agar lebih paham, kita perhatikan contoh code dibawah ini ya.



### Code!

```
var person = {
 firstName: "Kevin",
 lastName: "Dexsyah",
}

var school = {
 firstName: "Timedoar",
 lastName: "Academy",
}

function fullName () {
 return this.firstName + " " + this.lastName;
}

console.log(fullName.call(person));
console.log(fullName.apply(school));
```



Apa yang akan ditampilkan pada console?  
Bisakah kamu menjelaskan algoritma code diatas?



Sekarang, coba buat dua buah object dan sebuah function, lalu gunakan keyword `call()` dan `apply()`

Tunjukkan ke gurumu untuk diperiksa ya.



### Write here!

Untuk mengecek apakah code yang kamu tulis diatas benar, coba tulis kembali code mu di <https://playcode.io/>. Dan tunjukkan ke gurumu!

## Pertemuan 12- This is Object

### CHECK YOUR UNDERSTANDING



Hari/Tanggal : \_\_\_\_\_

1.

```
var movies = {
 title: "Star Wars: The Rise of Skywalker",
 releaseDate: "December 16, 2019",
 genre: "Science fiction"
};
```

Dari coding diatas yang mana termasuk nama objek? Apa saja propertiesnya?

.....

.....

.....

2.

Dalam coding no 1 diatas, bagaimana sintaks untuk mengakses nilai dari properties title?

.....

.....

.....

3.

```
var name = "Andra";

function mySelf(){
 name="Ani";
}
console.log(name)
```

Dalam coding diatas, apa yang akan ditampilkan ke dalam console?

.....

.....

.....



13

## CURRATED, OBJECT METHOD AND CREATE CANVAS



### Apa Yang Akan Kita Pelajari?

1. Mengetahui tentang currated live di sosial media.
2. Mengetahui konsep object method dalam coding
3. Mengetahui tentang canvas dalam JavaScript
4. Menggambar di canvas menggunakan JavaScript



# OBJECT METHOD AND CREATE CANVAS



## Warm Up!

- ❓ Masih ingat tidak dengan **object**?  
Apa saja nilai yang ditampung dalam sebuah object?

Yuk coba kita lihat lagi analogi object dalam mobil ini.

type : Tesla  
model : Model 3  
color : white

Start()  
Stop()  
Brake()  
Drive()

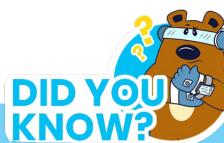


Object : Car

- ❓ Dalam object car, mana yang termasuk **properties**?  
Bagian mana termasuk **method**?

Hari ini kita akan belajar lebih banyak tentang **method**

- ❓ Masih ingat tidak apa itu **function**?



Properties sebuah object juga bisa berupa function loh!  
Properties itu lalu disebut sebagai method.



## What is Object Method?



### Vocabulary

#### Method

Method adalah perilaku dari objek yang berbentuk function.



Bagaimana **caranya** menulis method dalam object ya?

### Syntax!

```
var objectName = {
 methodName: function(){
 //command
 }
}
```

Sama seperti properties, **tulis dulu nama methodnya** kemudian diguki dengan **titik dua** `:` kemudian dikuti **keyword function()**, kamu bisa membuat function apapun yang ditulis dalam **curly bracket {}**



### Example

```
var car = {
 type:"Tesla",
 model:"Model 3",
 color: "White",
 sentence:function(){
 return "I have "+car.color+car.type;
 }
};
```



Dapatkah kamu menemukan **yang mana method?**  
Apa yang dilakukan oleh method tersebut?



## Pertemuan 13- Object Method and Create Canvas

### WHAT IS OBJECT METHOD?

Dalam contoh diatas, terdapat **method** yang bernama **sentence**. Method sentence ini digunakan untuk **membuat kalimat** I have white tesla.

 Sudah bisa **membedakan method dan properties?**



**Let's Practice!**

Coba tuliskan **object person** yang memiliki properties dan value seperti dibawah ini serta memiliki method **greeting** untuk mengucapkan Hi, I'm John

Name : John  
Age : 15  
Gender : Male  
Interest : Coding



Hi, I'm John

Tulis code mu disini ya!



**Write here!**

Untuk mengecek apakah code yang kamu tulis diatas benar, coba tulis kembali code mu di <https://playcode.io/>. Dan tunjukkan ke gurumu!



## Accessing Object Method



Lalu bagaimana caranya mengambil value dari method yang kita buat?

### Syntax!

```
objectName.methodName();
```

Hampir mirip dengan cara mengakses properties, hanya saja perlu diikuti **tanda kurung** buka tutup setelah nama method.



### Example

```
car.sentence();
// to print in console
console.log(car.sentence());
```



### Let's Practice!

Lanjutkan latihan tadi! Coba tuliskan cara memanggil **method sentence** dalam **object person** dan tampilkan ke dalam console.



### Write here!

Untuk mengecek apakah code yang kamu tulis diatas benar, coba tulis kembali code mu di <https://playcode.io/>. Dan tunjukkan ke gurumu!



## Using `this` Keyword



Apakah kamu memperhatikan dalam method yang kita buat, kita mengakses value dari object properties seperti ini.



### Example

```
var car = {
 type:"Tesla",
 model:"Model 3",
 color: "White",
 sentence:function(){
 return "I have "+car.color+car.type;
 }
};
```

Untuk mengakses value properties dalam sebuah method kita bisa menggunakan keyword `this`.

Sehingga codingnya menjadi seperti ini :



### Example

```
var car = {
 type:"Tesla",
 model:"Model 3",
 color: "White",
 sentence:function(){
 return "I have "+this.color+this.type;
 }
};
```



**Let's Practice!**

Lanjutkan latihan tadi! Coba buat **satu method** lagi yang mengakses value properties menggunakan keyword this. Tampilkan hasilnya dalam console!



**Write here!**

Untuk mengecek apakah code yang kamu tulis diatas benar, coba tulis kembali code mu di <https://playcode.io/>. Dan tunjukkan ke gurumu!



## 1. Create the Canvas



Pernahkah kalian **melukis atau menggambar?**

Media apa yang kalian gunakan untuk menggambar? Kertas atau Kanvas?



DID YOU KNOW?

Untuk membuat sebuah game dalam JavaScript juga memerlukan kanvas untuk menampung elemen-elemen game.



### Vocabulary

#### Canvas

Salah satu tag HTML yang berguna untuk menggambar dengan Syntax JavaScript di sebuah halaman web.

Kita akan menggunakan **HTML untuk membuat canvas**, tetapi untuk mengisi / **menggambar di canvas dilakukan dengan JavaScript**.

Yuk kita mulai **menggambar dengan JavaScript!**



Masih ingat tidak elemen HTML? Ada `<html>`, `<head>`, `<body>`, dan `<script>`.



## 1.1 Set up Canvas

```
<body>
 <canvas></canvas>
</body>
```

Pertama pastinya kita harus **membuat canvasnya** terlebih dahulu.

**Canvas dibuat diantara tag** `<body>` dan `</body>` dengan **menggunakan tag** `<canvas>`

### Syntax!

```
<canvas id=".." width=".." height="..></canvas>
```

Atribut `width` dan `height` diisi dengan nilai ukuran canvas yang ingin dibuat!

Kita juga bisa **menambahkan warna** ke dalam canvas dengan menambahkan atribut ini :

### Syntax!

```
style = "background-color:....;"
```

`background-color` diisi dengan nama warna yang diinginkan, misalnya `green`, `pink`.



Let's Practice!

**Yuk kita mulai!**

## 1. Preparation

1. Buka **Visual Studio**
2. Klik **File -> New File**
3. **Simpan file** dengan cara tekan Ctrl + S
4. Berikan nama "**Create Canvas By Namamu**"
5. Pilih **Save as type HTML** lalu klik **Save**

## 2. Coding

Yuk coba buat canvas sesuai dengan yang sudah dijelaskan diatas! Atau kamu bisa mengikuti code di bawah ini!



### Codes

```
<html>
<body>

<canvas id="myCanvas" width="300" height="100"
style="background-color : pink"></canvas>

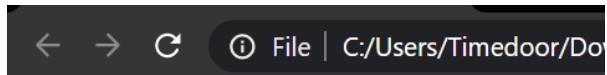
</body>
<script>
</script>
</html>
```

Code diatas akan membuat canvas dengan ukuran **lebar 300** dan **tinggi 100 berwarna pink**.



### 3. Run Program

Jangan lupa **Save ( Ctrl + S )** terlebih dahulu.  
Untuk menjalankan program gunakan **shortcut Alt + B!**



**Jika berhasil, di  
browser kalian akan  
tampil seperti ini!**



**Apakah kalian merasa Canvas yang dibuat terlalu kecil?**

Yuk ganti ukuran Canvas-mu, ikuti challenge dibawah ini!!!



**Challenge!**

**Yuk perbesar ukuran Canvas-mu.**

Ikuti petunjuk dibawah!

1. Ganti width (lebar) Canvas menjadi 640.
2. Ganti height (tinggi) Canvas menjadi 380.
3. Ganti warna Canvas sesuai yang kamu inginkan.



## ACCES THE CANVAS ELEMENT

### 1.2 Acces the Canvas Element

#### Canvas sudah berhasil kamu buat!

Untuk menggambar atau mengolah Canvas kita akan menggunakan JavaScript Syntax.

Nah, agar kamu bisa mengolah Canvas, kamu perlu **mengakses Elemen Canvas** terlebih dahulu.



**!** Karena kita akan membuat codes JavaScript , mulai dari code ini, ditulis diantara tag `<script>` dan `</script>`

 Tahukah kamu bagaimana cara mengakses Elemen Canvas?

Elemen Canvas dipanggil melalui atribut `id` yang telah dibuat pada tag `<canvas>` tadi. Yuk lihat code dibawah ini!

#### Syntax!

```
var varName= document.getElementById("canvasName");
```



#### Example

```
var canvas= document.getElementById("myCanvas");
```

Dalam contoh diatas, **canvas** yang dibuat yang bernama `canvas`.



## 2. Fill the Canvas Context



Canvas telah berhasil dipanggil. Sekarang kamu bisa mengisi Canvas dengan context sesuai keinginanmu! Apakah kalian tahu apa itu Context?



### Vocabulary

#### Context

Isi dari Canvas, berupa text, garis, lingkaran, dan gambar 2 dimensi.



Kita akan mengisi canvas dengan context bagian-bagian wajah seperti mata, hidung, dan mulut.



## 2.1 Acces the Canvas Context

Akses Canvas Context dengan code dibawah ini!

### Syntax!

```
var variableName = canvasName.getContext("2d");
```

`variableName` diganti dengan nama variable misalnya ctx atau context dan lainnya.  
`canvasName` diganti dengan nama variable dari canvas .  
`"2d"` artinya mengisi canvas dengan objek 2 dimensi.



Let's Practice!

## Yuk Lanjutkan Coding-mu!

Yuk coba akses canvas dan context sesuai dengan yang sudah dijelaskan diatas! Atau kamu bisa menambahkan coding-mu menjadi seperti ini!



### Codes

```
<html>
<body>

<canvas id="myCanvas" width="300" height="100"
style="background-color : pink"></canvas>

</body>
<script>
 var canvas = document.getElementById("myCanvas");
 var ctx = canvas.getContext("2d");
</script>
</html>
```

Jangan lupa **Save ( Ctrl + S )** terlebih dahulu.  
Untuk menjalankan program gunakan **shortcut Alt + B!**

 Apakah kamu **melihat perbedaan di Canvas-mu** setelah me-reload code sebelumnya?  
**Menurutmu**, mengapa hal itu bisa terjadi?

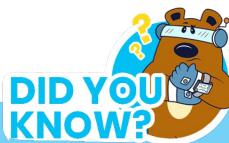


## 2.2 Make Some Texts



Kamu bisa mengisi Canvas dengan text loh!  
Tahukan kamu cara membuat text pada Canvas?  
Yuk perhatikan tabel dibawah!

Codes	Deskripsi
<pre>ctx.fillText("text", x, y);</pre> <p><b>Keterangan:</b> x = posisi koordinat x (width) y = posisi koordinat y (height)</p>	Membuat teks dengan <b>warna di posisi tertentu.</b> <b>Contoh:</b> <b>Hello World</b>
<pre>ctx.strokeText("text", x, y);</pre> <p><b>Keterangan:</b> x = posisi koordinat x (width) y = posisi koordinat y (height)</p>	Membuat teks <b>dengan outline</b> di posisi tertentu. <b>Contoh:</b> <b>Hello World</b>
<pre>ctx.font = "... px fontName";</pre>	Menentukan <b>ukuran font</b> dan <b>jenis font</b> . Titik-titik dapat diisi dengan <b>angka</b> dan dilanjutkan dengan <b>nama font</b> .
<pre>ctx.textAlign = "alignName";</pre>	Menentukan <b>rata kanan-kiri teks</b> . Value dari textAlign ("alignName") dapat diganti dengan "start", "end", "left", "right", atau "center".



JavaScript akan mengeksekusi kode setiap baris dari atas ke bawah.

Sehingga, penulisan code `ctx.font` dan `ctx.textAlign` harus didefinisikan terlebih dahulu sebelum kamu membuat text dengan `ctx.fillText` atau `ctx.strokeText`.



### Let's Practice!

## Yuk Lanjutkan Coding-mu!

**Yuk coba buat text berisi "Welcome" pada bagian atas Canvas-mu!**

Tambahkan coding-mu dalam tag `<script></script>` menjadi seperti ini!



### Codes

```
<script>
 var canvas = document.getElementById("myCanvas");
 var ctx = canvas.getContext("2d");

 // Wellcome Text
 ctx.textAlign = "center";
 ctx.font = "40px cursive";
 ctx.fillText("Welcome", 325, 60);

</script>
```

Jangan lupa **Save ( Ctrl + S )** terlebih dahulu.  
Untuk menjalankan program gunakan **shortcut Alt + B!**



← → ⌂ File | C:/Users/Timedoor/Desktop/testing.html



Apakah canvas  
mu sudah terlihat  
seperti gambar  
disamping ini?

Welcome



### Challenge!

Buat **stroke text** dibawahnya berisi "**This is My Canvas**" dengan **code strokeText()**;  
Gunakan font serif berukuran **20px**, pada posisi **x= 325, y=100**.

← → ⌂ File | C:/Users/Timedoor/Desktop/testing.html

Welcome

This is my Canvas

Setelah  
mengerjakan  
challenge,  
canvasmu akan  
terlihat seperti ini!



## Let's Explore

Coba ganti `textAlign` dengan value lain seperti "start", "end", "left", dan "right"!

Tulisakan perbedaan dari 5 value `textAlign` yang sudah kamu coba!



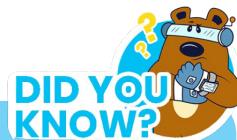
## 2.3 Coloring Text



Bosan nggak sih kalau text yang kamu buat warna hitam aja?

Kamu bisa **ganti warna** lain loh! Caranya ada di tabel ini!

Codes	Deskripsi
<code>ctx.fillStyle = "color";</code>	<b>Memberikan warna</b> pada suatu bidang. Warnanya harus ditulis dalam Bahasa Inggris, contohnya "pink", "blue", "orange", "red", "green".
<code>ctx.strokeStyle = "color";</code>	Memberikan <b>warna pada garis luar (outline)</b> bidang. Cara penulisan warna sama seperti pada kode <code>fillStyle</code>



Sama seperti penulisan code `ctx.font` dan `ctx.TextAlign`. Code style (`ctx.fillStyle` dan `ctx.strokeStyle`) harus ditulis terlebih dahulu sebelum `ctx.fillText` atau `ctx.strokeText`!



Let's Practice!

## Yuk Lanjutkan Coding-mu!

**Yuk coba buat text "Welcome" berwarna merah!**

Tambahkan coding-mu dalam tag <script></script> menjadi seperti ini!



Codes

```
<script>
 var canvas = document.getElementById("myCanvas");
 var ctx = canvas.getContext("2d");

 // Wellcome Text
 ctx.fillStyle = "red";
 ctx.textAlign = "center";
 ctx.font = "40px cursive";
 ctx.fillText("Welcome", 325, 60);
</script>
```



**Save ( Ctrl + S )** terlebih dahulu.

Reload tab Create Canvas pada browser **(Ctrl + R.)**

**Jika berhasil, program akan terlihat seperti berikut ini.**



## Let's Explore

Coba letakkan kode `ctx.fillStyle` setelah `ctx.fillText` dan lihat apa yang terjadi!  
Menurutmu, mengapa hal itu bisa terjadi?



## 2.4 Drawing Rectangles

Selain text, kamu juga bisa menggambar persegi pada Canvas-mu loh!  
Tahukan kamu cara menggambar rectangles atau persegi pada Canvas?

Codes	Deskripsi
<code>ctx.fillRect(x, y, width, height);</code>	Menggambar persegi (rectangles) dengan mengisi warna di bidangnya. 
<code>ctx.strokeRect(x, y, width, height);</code>	Menggambar sisi berbentuk persegi (rectangle). 
<code>ctx.clearRect(x, y, width, height);</code>	Menghapus sebuah bidang dengan bentuk persegi (rectangle). 

### Keterangan:

x = posisi koordinat x  
y = posisi koordinat y

width = lebar persegi (rectangle)  
height = tinggi persegi (rectangle)



Kamu juga bisa **memberikan warna pada rectangle** / persegi yang kamu buat loh!

Cara dan aturannya sama dengan kamu memberikan warna pada Text, yaitu menggunakan `ctx.fillStyle / ctx.strokeStyle` dan ditulis sebelum kode pembuatan rectangle.



### Let's Practice!

## Yuk Lanjutkan Coding-mu!

**Yuk coba buat persegi di bagian tengah bawah sebagai mulut.**  
Tambahkan coding-mu dalam tag `<script></script>` menjadi seperti ini!



### Codes

```
<script>
 ctx.fillStyle = "red";
 ctx.textAlign = "center";
 ctx.font = "40px cursive";
 ctx.fillText("Welcome", 325, 60);

 // rectangle
 ctx.fillStyle="red";
 ctx.fillRect(200, 325, 250, 50);

</script>
```

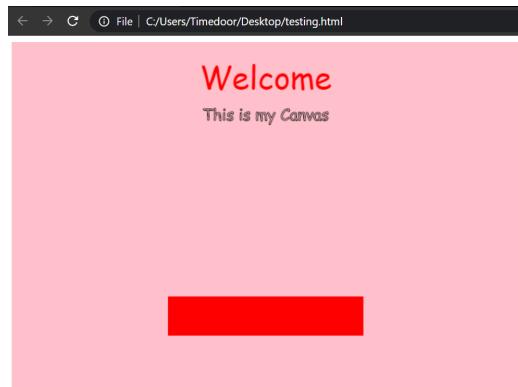


## Pertemuan 13- Object Method and Create Canvas

### DRAWING RECTANGLES

**Save ( Ctrl + S )** terlebih dahulu.  
Reload tab Create Canvas pada browser **(Ctrl + R.)**

Jika berhasil, program akan terlihat seperti berikut ini.



#### Challenge!

- Coba gunakan code `clearRect();`
- Atur `x = 30, y = 30, width = 80, height = 40.`
- Reload dan lihat hasilnya!





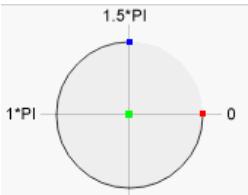
## 2.4 Drawing Path



Kamu juga bisa **menggambar bentuk lain** sesuka kamu.

### Bentuk apa aja sih kira-kira yang bisa digambar?

Kamu bisa menggambar segitiga, trapesium, bahkan lingkaran. Namun, untuk menggambar bentuk-bentuk itu, kamu membutuhkan sebuah **Path**.

Codes	Deskripsi
<code>ctx.beginPath();</code>	Membuat path baru dan mengakhiri path lain.
<code>ctx.closePath();</code>	Memindahkan pena ke titik awal pada path
<code>ctx.lineTo(x, y);</code>	Memindahkan pena ke titik selanjutnya.
<code>ctx.moveTo(x, y);</code>	Memposisikan titik awal sebuah path.
<code>ctx.arc(x, y, r, startAngle, endAngle, anticlockwise);</code>	<p>Menggambar sebuah busur, biasanya digunakan untuk menggambar lingkaran / setengah lingkaran.</p> <p><b>Keterangan:</b></p> <ul style="list-style-type: none"> <li><code>arc</code> : kurva atau lengkung</li> <li><code>x, y</code> = titik tengah lingkaran</li> <li><code>r</code> = jari-jari lingkaran</li> <li><code>startAngle</code> = sudut awal, pada lingkaran penuh set <code>startAngle = 0</code>.</li> <li><code>endAngle</code> = sudut akhir, pada lingkaran penuh set <code>endAngle = 2*Math.PI</code></li> <li><code>anticlockwise</code> = berisi true dan false, true berarti tidak searah jarum jam, sedangkan false berarti searah jarum jam.</li> </ul>
*Perhatikan titik berwarna dan lihat keterangannya disamping!	



## DRAWING PATH

Codes	Deskripsi
<code>ctx.stroke();</code>	Menampilkan path yang dibuat dalam <b>bentuk garis</b> , warna default adalah hitam. Untuk memberikan warna lain, atur warna dengan <code>ctx.strokeStyle</code>
<code>ctx.fill();</code>	Menampilkan path yang dibuat dalam <b>bentuk bidang</b> , warna default adalah hitam. Untuk memberikan warna lain, atur warna dengan <code>ctx.fillStyle</code>



## Let's Practice!

## Yuk Lanjutkan Coding-mu!

**Yuk coba buat triangle sebagai hidungnya!**

Tambahkan coding-mu dalam tag `<script></script>` dengan code dibawah ini!



## Codes

```

<script>
 // make a triangle
 ctx.fillStyle="red";
 ctx.beginPath();
 ctx.moveTo(325, 230);
 ctx.lineTo(300, 270);
 ctx.lineTo(350, 270);
 ctx.closePath();
 ctx.fill();
</script>

```



← → ⌂ File C:/Users/Timedoar/Desktop/testing.html

**Save ( Ctrl + S )** terlebih dahulu.

Reload tab Create Canvas pada browser **(Ctrl + R.)**

Jika berhasil, program akan terlihat seperti berikut ini.



## Yuk Lanjutkan Coding-mu!

Sekarang kita buat dua lingkaran sebagai mata.

Tambahkan coding-mu dalam tag `<script></script>` dengan code dibawah ini!



### Codes

```
<script>
 // make a circle
 ctx.fillStyle = "black";
 ctx.strokeStyle = "white";
 ctx.lineWidth = "20";
 ctx.beginPath();
 ctx.arc(130, 150, 40, 0, 2 * Math.PI);
 ctx.fill();
 ctx.stroke();
</script>
```

### DRAWING PATH



**Save ( Ctrl + S )** terlebih dahulu.

Reload tab Create Canvas pada browser (**Ctrl + R.**)

Jika berhasil, program akan terlihat seperti berikut ini.



#### Challenge!

Bisakah kamu menambahkan satu lingkaran lagi sebagai mata kiri?

- Tambahkan satu lingkaran lagi dengan `fillStyle`, `strokeStyle`, `lineWidth` yang sama di posisi **x = 520, y = 150, r = 40, endAngle = 0, 2 \* Math.PI.**





Hari/Tanggal : \_\_\_\_\_

1. Apa itu method dalam object JavaScript?

.....  
.....  
.....

- 2.
- ```
ctx.strokeStyle = "green";
ctx.fillStyle = "red";
ctx.font = "60px cursive";
ctx.fillText("Welcome", 10, 60);
```

Menurutmu, warna apa yang akan muncul pada text "Welcome"?

.....
.....
.....

3. Menurutmu, mengapa kode style harus dituliskan sebelum kode inti seperti `fillText` dan `fillRect`?

.....
.....



Timedooring
Coding Academy

14

JAVASCRIPTS CLASSES AND SNAKE GAME



Apa Yang Akan Kita Pelajari?

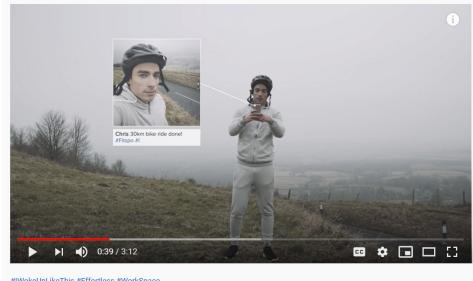
1. Mengetahui konsep class dalam JavaScript
2. Membuat game snake menggunakan object dan canvas



JAVASCRIPT CLASSES AND SNAKE GAME



Check This Video



#IwokeUpLikeThis #Effortless #WorkSpace
Are You Living an Insta Lie? Social Media Vs. Reality



Apa pendapatmu dari video ini?



Apakah kamu memiliki pengalaman yang sama?



Menurutmu, apa dampak positif dan negatif dari hal ini?



Terkadang, apa yang ditampilkan disosial media belum tentu hal yang sebenarnya. Seseorang bisa saja menjadi dua pribadi yang berbeda di real life dan di sosial media. Hal itu terjadi karena dengan sosial media, kita punya kesempatan untuk mengatur apa yang kita bagikan dengan orang lain. Jadi kita tidak perlu iri dengan apa yang kita lihat di sosial media.



Currate

Memilih, mengatur, dan menjaga foto, video, dll



Currate merupakan Hal yang biasanya dilakukan ketika akan membagikan post di sosial media.



Bagaimana dengan mu?



Seberapa besar kamu merasa dipengaruhi oleh apa yang kamu lihat secara online?



Seberapa banyak jumlah post-mu yang benar-benar mewakili dirimu?



Ada juga istilah **Red Flag Feeling**. Jika kamu melihat sosial media lalu kamu merasa tidak nyaman, sedih, dsb itu artinya kamu harus **aware** dan **STOP** sosial media sementara.

Meskipun kita punya kesempatan untuk memilih apa yang ingin kita tampilkan di sosial media, tetapi harus tetap ingat untuk **Pause and Think sebelum Posting**. Karena apa yang akan kita tampilkan akan dilihat dan dinilai oleh orang lain.



Warm Up!



Masih ingat tidak tentang objek?

Bagaimana cara membuat objek dalam JavaScript?



Bagaimana jika memiliki **3 objek** yang memiliki **bentuk yang sama**, hanya **berbeda value** di setiap **property atau methodnya**?



Bagaimana cara menulisnya dalam JavaScript?



Example

```
var car1= {  
    type: "Tesla",  
    color: "White"  
};  
var car2= {  
    type: "Mercedes",  
    color: "Black"  
};  
var car3= {  
    type: "BMW",  
    color: "Gray"  
};
```



Apakah menurutmu bisa ditulis seperti ini?

Code disamping bukan solusi yang terbaik!

Ini akan menjadi sebuah masalah bila kita di suruh membuat objek yang jumlahnya puluhan atau ratusan.



Lalu bagaimana caranya?

Caranya adalah dengan **membuat template** atau istilahnya **blueprint dari objek** tersebut.

Di javascript ada 2 cara untuk membuat template dari sebuah objek. Pertama dengan menggunakan **function constructor** dan yang kedua menggunakan **class**.



1. Function Constructor

Function constructor adalah sebuah function declaration biasa. Untuk membedakan function contructor dengan function declaration biasa adalah dengan memakai **huruf besar pada kata pertama** nama function tersebut.

Perhatikan penulisan function constructor dibawah ini!

Syntax!

```
//make template
function FunctionName(properties, method) {
    this.properties = properties;
    this.method = function() {
        //function command
    };
}
//create new object using template
var objectName = new FunctionName(properties, method);
```



Example

```
function Car(type, color){
    this.type=type;
    this.color=color;
    this.display = function(){
        document.write(this.color + " " + this.type)
    }
}

var car1= new Car("Tesla", "White");
var car2= new Car("BMW", "Grey");
var car3= new Car("Mercedes", "Black");

car1.display();
```

Pertemuan 14- JavaScript Classes and Snake Game

JAVASCRIPT CLASS



Untuk **membuat objek dari template** yang sudah dibuat digunakan **keyword new** lalu diikuti dengan **pemanggilan function car** yang berisi **value dari object properties**.



2. Class



Menurutmu, **apa itu class?**

Sama seperti Function constructor, **Class adalah sebuah template untuk membuat objek**



Class adalah salah satu fitur baru yang di perkenalkan di javascript versi 2015 atau ES6. Fitur class menyediakan syntax yang lebih sederhana dan rapi.

Perhatikan penulisan function constructor dibawah ini!

Syntax!

```
//make a template
class ClassName {
  constructor() {
    this.properties = properties;
    this.method = function(){
      //command
    }
  }
}
//create new object using template
var objectName = new ClassName(properties, method);

//accesing object properties
objectName.properties;
// accesing object method
objectName.method();
```



Untuk membuat 3 objek mobil tadi, kita bisa menggunakan class seperti code dibawah ini!



Example

```
//create class
class Car {
    constructor(type, color) {
        this.type = type;
        this.color = color;
        this.display = function(){
            document.write(this.color + " " + this.type + "<br>")
        }
    }
}
//create object
var car1 = new Car("Tesla", "White");
var car2 = new Car("BMW", "Grey");
```

Untuk membuat objek dari class caranya sama seperti membuat objek dari function constructor!

Untuk lebih jelas, coba perhatikan gambar ini!

Class : Car



Object 3

Type : Mercedes
Color : Black

Object 2

Type : BMW
Color : Grey

Object 1

Type : Tesla
Color : White

Pertemuan 14- JavaScript Classes and Snake Game

JAVASCRIPT CLASS



Coba kerjakan contoh diatas, lalu akses method display dari masing-masing object! Sehingga muncul seperti gambar disamping!

white Tesla
black BMW



Challenge!

Coba tuliskan **class Student** yang memiliki **properties; name dan hobby**. Dari class tersebut lalu **buat 5 object** yang memiliki **properties value** seperti tabel dibawah ini! object juga memiliki method yang **display** untuk menampilkan kalimat : *I am I like*

Kerjakan di <https://playcode.io/>

Class : Student

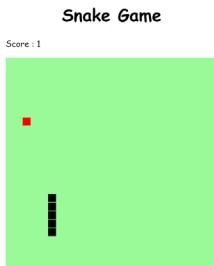


| object | Name | Hobby |
|----------|--------|------------|
| Student1 | Andra | Reading |
| Student2 | Bella | Cooking |
| Student3 | Caka | Gaming |
| Student4 | Davina | Coding |
| Student5 | Evan | Basketball |



Project

Snake Game



Pernahkah kamu memainkan snake game?

Kali ini kita akan membuat snake game dengan JavaScript.

Dalam game ini, snake harus memakan apple sebanyak-banyaknya.

<https://timedoorsnakegame.netlify.app/>



Masih ingat kan tentang **Canvas dan Object**?

Kita akan menggunakan konsep Canvas dan Object dalam game ini.

1. Prepare Template

1. Buka link di browser <https://github.com/timedoorsnakegame-template.git>
2. Extract file yang di download kemudian buka folder template di **Visual Studio Code**

2. Coding

Sekarang, dalam file **index.html** tuliskan code berikut ini.

Pertemuan 14- JavaScript Classes and Snake Game

SNAKE GAME



Step 1. Create Canvas

Ayo kita buat dulu canvasnya sebagai tempat elemen gamenya
Tambahkan code ini pada tag `<canvas>...</canvas>`

```
<body>
  <h1>Snake Game</h1>
  <p>Score : <span id="score"> </span></p>

  <canvas></canvas>

</body>
```

Pada bagian ini,
tambahkan code
untuk mengatur
lebar, tinggi, dan id
dari canvas.



Codes

```
<canvas width="400" height="400" id="game"
style="background-color : palegreen" ></canvas>
```

Code diatas akan membuat canvas dengan ukuran tinggi dan panjang 400 piksel dengan warna palegreen.

Kamu bisa mengganti warnanya dengan yang kamu inginkan.

Step 2. Acces Canvas & Context



Karena kita akan membuat codes JavaScript , mulai dari code ini, ditulis diantara tag `<script>` dan `</script>`

```
<body>
  <h1>Snake Game</h1>
  <p>Score : <span id="score"> </span></p>

  <canvas width="400" height="400" id="game" style="background-color : palegreen" ></canvas>

</body>

<script>
</script>
```



Tambahkan code ini pada tag `<script></script>` dibawah komentar Step 2.

```
<script>
    // Step 2. Acces Canvas & Context
```



Codes

```
var canvas= document.getElementById("game");
var ctx = canvas.getContext("2d");
```

Code diatas digunakan untuk mengakses elemen canvas dan mengakses elemen canvas untuk diisi context 2d.

Step 3. Initialize Variable

Lanjutkan coding pada tag `<script></script>` dibawah komentar Step 3.

```
// Step 3. Initialize Variable
```



Codes

```
var score=0;
var count =0;
var grid = 16;
```

Dalam code diatas kita membuat 3 variable.

Variable score digunakan untuk menampung nilai score

Variable count digunakan untuk menghitung frame per second sebagai kecepatan dari snake.

Variable grid digunakan untuk mengatur grid menjadi 16 pixel.



Step 4. Create Object

Sekarang kita buat object snake dan apple .

Tambahkan code ini pada tag `<canvas>...</canvas>` dibawah komentar Step 4.

```
// Step 4. Create object
```



Codes

```
// create object snake
var snake = {
    x: 160,
    y: 160,
    dx: grid,
    dy: 0,
    cells: [],
    maxCells: 4
};

//create object apple

var apple = {
    x: 320,
    y: 320
};
```

Dalam code diatas kita membuat 2 Object.

Object snake berisi properties x untuk koordinat x, y untuk koordinat y, dx untuk kecepatan horizontal, dy untuk kecepatan vertical, cells untuk panjang snake, dan maxCell

Object apple berisi properties x untuk koordinat x, y untuk koordinat y.



Step 5. Key Arrow Controller

Sekarang kita buat agar snake bergerak mengikuti key arrow yang ditekan.

Tambahkan code ini pada tag <canvas>...</canvas> dibawah komentar Step 5.

```
// step 5. Key Arrow Controller
document.addEventListener('keydown', function(e) {  
});
```



Codes

```
document.addEventListener('keydown', function(e) {
  // left arrow key
  if (e.which === 37 && snake.dx === 0) {
    snake.dx = -grid;
    snake.dy = 0;
  }
  // up arrow key
  else if (e.which === 38 && snake.dy === 0) {
    snake.dy = -grid;
    snake.dx = 0;
  }
  // right arrow key
  else if (e.which === 39 && snake.dx === 0) {
    snake.dx = grid;
    snake.dy = 0;
  }
  // down arrow key
  else if (e.which === 40 && snake.dy === 0) {
    snake.dy = grid;
    snake.dx = 0;
  }
});
```



3. Save

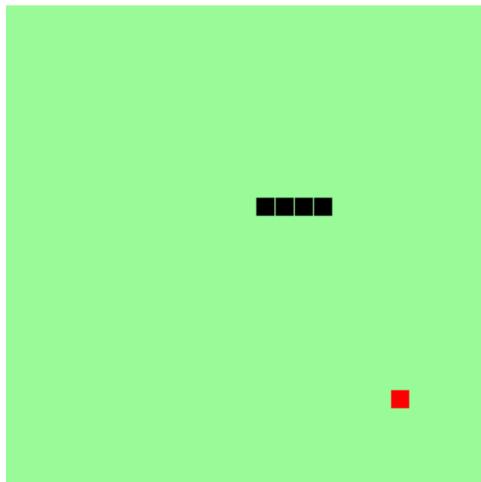
Simpan file dengan **klik file** lalu pilih **Save**.

4. Run Code

Sekarang coba **jalankan program di browsermu** dengan menekan **Alt+B** dan coba test apakah game mu berjalan dengan baik?

Snake Game

Score :



Pertemuan 14- JavaScript Classes and Snake Game

CHECK YOUR UNDERSTANDING



Hari/Tanggal : _____

1. Mengapa class di butuhkan dalam membuat object ?

.....
.....
.....

- 2.
- ```
class Game {
 constructor(name, genre) {
 this.name=name;
 this.genre=genre;
 }
}
```

Dari class diatas, bagaimana cara membuat object yang bernama game1 dengan properties name="Dota 2", genre="MOBA" ?

.....  
.....  
.....

3. Dalam snake game, canvas digunakan untuk apa?

.....  
.....  
.....

4. Dalam snake game, object digunakan untuk apa?

.....  
.....  
.....



# 15

## ONLINE HATE SPEECH & DEPLOY



### Apa Yang Akan Kita Pelajari?

1. Mengetahui konsep object method dalam coding
2. Mengetahui tentang canvas dalam JavaScript
3. Menggambar di canvas menggunakan JavaScript



# ONLINE HATE SPEECH & DEPLOY



## Warm Up!

Pernahkah kamu mendengar istilah **online hate speech**?

Pernahkah kamu menemukan **online hate speech? dimana?**



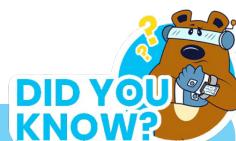
## Vocabulary

### Hate Speech

Serangan verbal yang menargetkan seseorang karena kelompoknya, misalnya karena RAS, agama, jenis kelamin, dll.



Menurutmu, kenapa seseorang membuat dan menyebarkan **hate speech**?



Bukan tanpa alasan atau sekadar iseng seseorang membuat dan menyebarkan hate speech, faktor utama yang menyebabkannya adalah prasangka buruk terhadap orang atau kelompok tertentu.

Sering kali, seseorang melakukannya secara anonymous atau menggunakan fake account.



Pernahkah kamu mendengar tentang **kasus perusakan bendera merah putih** di Surabaya yang diduga dilakukan mahasiswa Papua?

NATIONAL GEOGRAPHIC INDONESIA  
BERITA FOTO LEPAS MAJALAH INDONESIA 360 JURNAL XPLORASI

Dugaan Perusakan Terhadap Bendera Merah Putih oleh Oknum Mahasiswa Papua di Surabaya, Berikut Kronologisnya...

Mahrud Zulfikar - Senin, 19 Agustus 2019 | 18:29 WIB



Kenapa kemudian orang mudah **percaya dengan dugaan** tersebut sehingga menjadi konflik?

**Ini terjadi karena bias.** Mungkin saja orang memiliki bias bahwa orang papua adalah orang yang kasar, sehingga saat melihat berita yang tersebar langsung marah, tanpa mengecek dahulu kebenarannya.



### Vocabulary

#### Confirmation Bias

Kecenderungan untuk menafsirkan informasi dengan cara yang menegaskan apa yang sudah kita yakini

#### Cognitive Bias

Keterbatasan dalam menafsirkan informasi yang disebabkan oleh pemahaman informasi hanya melalui pengalaman dan preferensi pribadi.



Bagaimana cara menghadapi hate speech

## Tips Responding Online Hate Speech

1.

### Speak Your Thruth

Katakan perasaan, pemikiran dan persepsi mu secara jujur menegenai suatu masalah

2.

### Respect Others Viewpoint

Cari juga sumber lain yang berbeda point view denganmu



3.

### Don't Interupt

Tidak menyela artinya kamu mendengarkan dan memperhatikan pendapat orang lain.

4.

### Disagree Respectfully

Tidak masalah jika kamu memiliki pendapat yang berbeda, tapi sampaikan dengan cara yang sopan.



## Deploy Your Game



Di pertemuan sebelumnya kita sudah menyelesaikan snake game menggunakan JavaScript.  
Senangkah kamu bisa memainkan game-mu sendiri?



Tetapi game yang kamu buat hanya bisa dijalankan di komputermu saja.



Bagaimana agar kita bisa membagikannya ke teman?

Agar teman-temanmu bisa memainkan gamenya, kamu harus melakukan **deployment** terlebih dahulu!

Kali ini kita akan mendeploy snake game menggunakan netlify!



### Vocabulary

#### Deployment

Kegiatan yang membuat sebuah aplikasi siap untuk dijalankan oleh pengguna/ mem-publish sebuah aplikasi

#### Netlify

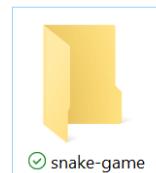
Layanan Software as a Service yang memberikan kemudahan untuk mempublish web statis.



**Yuk kita mulai!**

## 1. Prepare Source Code in a Folder

1. **Buat folder baru** di komputermu dan beri nama **snake-game**.

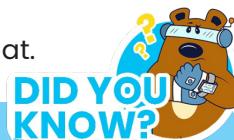


2. **Buka kembali coding JavaScript snake game** sebelumnya. Kemudian pilih File lalu klik Save As dan beri nama "**index.html**".

**Simpan di folder snake-game** yang baru kita buat.

Kenapa namanya harus "**index.html**"?

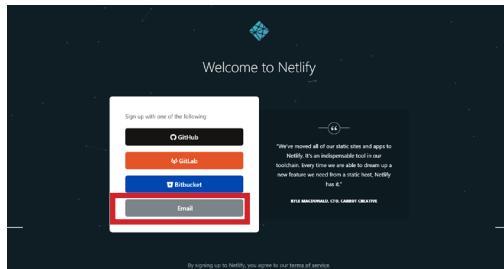
Karena file index berfungsi sebagai index dari sebuah website. Ketika melakukan deployment, server akan mencari file dengan nama index.html, jika ini tidak ada maka akan ditampilkan kode error.



## 2. Create Netlify Account

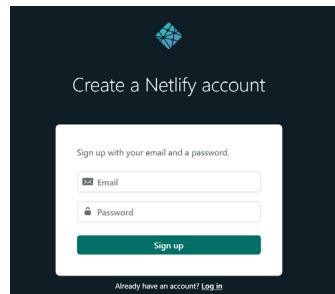
Untuk bisa mendeploy aplikasi di netlify kamu harus memiliki akun terlebih dahulu. Siapkan akun email dan password ya!

1. Buka <https://app.netlify.com/signup> di browsermu. Kemudian pilih Email





2. Buat akun dengan mengisi email dan password.

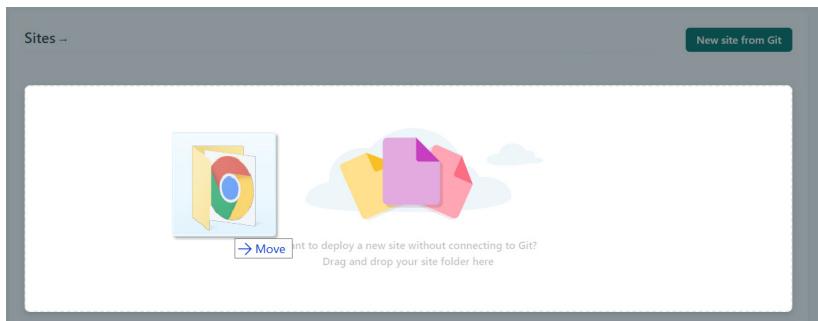


3. Buka emailmu dan lakukan verifikasi.



### 3. Deploy Your First Game

Setelah punya akun,yuk deploy game pertama-mu!  
Caranya : **Drag folder snake-game ke dalam kolom sites**



## Pertemuan 15-Online Hate Speech & Deploy

### DEPLOY YOUR GAME

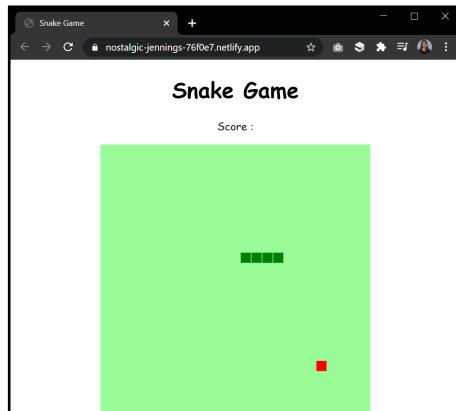


**Tunggu sampai aplikasimu berhasil di deploy.**

Jika berhasil akan muncul link dari game mu dengan warna hijau.

The screenshot shows the Netlify 'Site overview' page for a team named 'ayukrisnasari15's team'. It displays a deployment log entry for 'nostalgic-jennings-76f0e7' with a green link: <https://nostalgic-jennings-76f0e7.netlify.app>. Below the log, it says 'Manual deploys. Last published at 2:42 PM.' and shows two buttons: 'Site settings' and 'Domain settings'. At the bottom, there's a 'Getting started' section with three numbered steps: 1. 'Your site is deployed ✓' (with a note: 'Try a test build and deploy, directly from your Git repository or a folder.'), 2. 'Set up a custom domain' (with a note: 'Buy a new domain or setup a domain you already own.'), and 3. 'Secure your site with HTTPS' (with a note: 'Your site is secured automatically with a Let's Encrypt certificate.').

Klik link yang diberikan untuk menguji apakah game mu sudah berhasil di deploy!



#### 4. Change Your Site Name



Netlify memberikan nama link diberikan secara acak. Bagaimana jika kita ingin mengubahnya menjadi link yang mudah diingat?



### 1. Pada site overview klik **Site Setting**

A screenshot of a Netlify site overview page. It shows the site name "nostalgic-jennings-76f0e7" and its URL "https://nostalgic-jennings-76f0e7.netlify.app". Below this, it says "Manual deploys. Last published at 2:45 PM." At the bottom, there are two buttons: "Site settings" (highlighted with a red box) and "Domain settings".

### 2. Pada bagian Site Information, kik **change site names**

A screenshot of the "Site details" section of a site's configuration. It includes a "Site information" table with the following data:

|              |                                      |
|--------------|--------------------------------------|
| Site name:   | nostalgic-jennings-76f0e7            |
| Owner:       | ayukrisnasari15's team               |
| API ID:      | cc76d1a1-64a0-4284-af85-9c495bb9e507 |
| Created:     | Today at 2:42 PM                     |
| Last update: | Today at 2:45 PM                     |

At the bottom, there are two buttons: "Change site name" (highlighted with a red box) and "Transfer site ▾".

3. Kemudian **beri nama** sesuai dengan yang kamu ingin!  
Jika sudah selesai, klik **Save**.

A screenshot of a "Change site name" dialog box. It contains the following text:  
The site name determines the default URL for your site. Only alphanumeric characters and hyphens are allowed.  
Site name: snakegame-by-ayu  
https://snakegame-by-ayu.netlify.app

At the bottom are "Save" and "Cancel" buttons.

**Jika berhasil**, link game mu akan berubah sesuai dengan nama yang kamu berikan.  
Kamu bisa membagian link tersebut ke teman!



## Updating Your Game



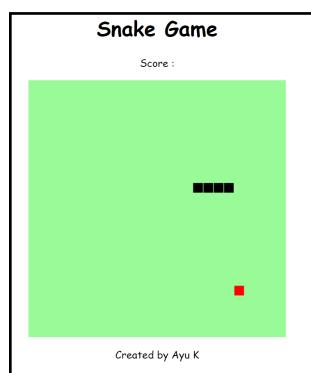
Seberapa sering game favoritmu memintamu untuk melakukan update?



Dalam **game development**, tentunya akan ada proses update.

Menurutmu saat dilakukan update, apakah programmer membuat ulang codingnya atau hanya menambahkan dari coding yang sebelumnya?

**Mari kita update snake game** dengan menambahkan kecepatan snake dan memberi namamu.



### 1. Coding Update

### 1. Open Project

Buka **file index.html** yang kamu simpan di folder snake-game di visual studio code.



## 2. Coding Update

### Step 1. Change snake speed

Perhatikan function loop. Pada game sebelumnya, kecepatan snake adalah 10 frame per second.  
 $( 60 : 6 = 10 )$ .

```
// game loop
function loop() {
 requestAnimationFrame(loop);
 if (++count < 6) {
 return;
 }
}
```

Sekarang ayo kita buat menjadi lebih cepat dengan **mengganti angka 6 menjadi angka 3**.

Ini akan membuat snake memiliki kecepatan 20 frame per second.

Semakin kecil angkanya,  
kecepatan snake semakin besar.

```
// game loop
function loop() {
 requestAnimationFrame(loop);
 if (++count < 3) {
 return;
 }
}
```

**Sekarang coba Save ( Ctrl + S ) lalu run program ( Alt +B ).  
Apakah snakenya berjalan lebih cepat?**



## Step 2. Add Your Name

Sekarang kita tambahkan namamu dibagian bawah canvas!.

Pada tag `<body>...</body>`, tepat dibawah tag `<canvas>` tambahkan code berikut ini



### Codes

```
<p> Created by </p>
```

Lengkapi dengan namamu ya!

```
<body>
 <h1>Snake Game</h1>
 <p>Score : </p>
 <!-- Step 1. Create Canvas-->
 <canvas width="400" height="400" id="ga

 <p>Created by Ayu K </p>

</body>
```

**Sekrang coba Save ( Ctrl + S ) lalu run program ( Alt +B ).  
Apakah namamu sudah muncul?**



### Challenge!

Kamu juga bisa membuat update lainnya pada gamemu.  
Misalnya mengganti warna canvas, mengganti warna snake dan lainnya.



## 2.Update Deployment



Coba buka link game mu yang telah di deploy.  
Apakah game-mu terupdate disana?

Biarpun kita sudah melakukan update pad coding, tetapi game yang terdeploy masih menggunakan coding sebelumnya. Jadi kita juga harus mengupdate deployment.

**Mari kita update deployment snake game.**

1. Buka netlify dan pilih site snake game. Setelah itu klik pada **Deploys**.

The screenshot shows the Netlify Site overview for the 'snakegame-by-ayu' site. A red box highlights the 'Deploys' tab. Below it, there's a summary card for the site, showing the URL <https://snakegame-by-ayu.netlify.app>, a manual deploy message, and a preview image of the Snake Game.

2. **Drag folder** snake-game yang telah di update ke dalam kolom deploy.  
Tunggu hingga deployment selesai.

The screenshot shows the Netlify Deploys page. A red box highlights the area where a new deployment is being added. It shows a placeholder message: 'Need to update your site? Drag and drop your site folder here'. Below this, two previous deployments are listed: one from 'Production' at 2:45 PM and another from 'Production' at 2:42 PM.



## Intro to Phaser Framework



Coba kita perhatikan lagi coding di snake game.

Untuk membuat snake bergerak mengikuti arrow key, kita menggunakan coding dibawah ini!

Kita harus tau kode dari masing-masing arrow key .



### Codes

```
document.addEventListener('keydown', function(e) {
 // left arrow key
 if (e.which === 37 && snake.dx === 0) {
 snake.dx = -grid;
 snake.dy = 0;
 }
 // up arrow key
 else if (e.which === 38 && snake.dy === 0) {
 snake.dy = -grid;
 snake.dx = 0;
 }
 // right arrow key
 else if (e.which === 39 && snake.dx === 0) {
 snake.dx = grid;
 snake.dy = 0;
 }
 // down arrow key
 else if (e.which === 40 && snake.dy === 0) {
 snake.dy = grid;
 snake.dx = 0;
 }
});
```



Untuk membuat snake memakan apple kita juga harus tau koordinat dari snake dan apple seperti ini.



## Codes

```
if (cell.x === apple.x && cell.y === apple.y) {

 snake.maxCells++;
 apple.x = getRandomInt(0, 25) * grid;
 apple.y = getRandomInt(0, 25) * grid;
}
```



Memusingkan dan sulit dibaca bukan?

Dengan bahasa **JavaScript native**, semua harus dijabarkan agar dimengerti oleh komputer.  
Tapi bagaimana kalau **ada library** berisi function, method, class dan properties yang sudah langsung bisa kita pakai?



Misalnya membuat key arrow controller dengan code ini?



## Codes

```
// key arrow controller
var isUpDown = cursorKeys.up.isDown ;
var isDownDown = cursorKeys.down.isDown ;
var isLeftDown = cursorKeys.left.isDown ;
var isRightDown = cursorKeys.right.isDown ;

//collision
game.physics.arcade. collide (ball, paddle);
```



Jadi lebih sederhana dan mudah untuk dibaca bukan?  
**Itulah gunanya Library.**



## JavaScript Library

Library dari pre-written JavaScript yang memudahkan development yang berbasis JavaScript

## Framework

Library yang sudah memiliki kerangka kerja.  
Contohnya : Phaser

## Phaser 3



Framework yang akan kita gunakan untuk membuat game adalah **Phaser 3**.

Phaser 3 adalah framework game 2D yang digunakan untuk membuat game HTML5 untuk desktop dan mobile..

**Kita akan belajar menciptakan banyak games menggunakan Phaser 3 di Intermediate 2.**



**Next meeting will be exam!**



Bagaimana belajar JavaScript codingnya?

Untuk mengakhiri Intermediate 1, Siapkan presentasi dengan materi yang akan diacak oleh guru!

Presentasi 3-5 menit yang berisi :

- Perkenalan
- JavaScript syntax (misal : syntax array)
- Contoh penggunaannya ( contoh coding array)
- Ucapan terimakasih



Hari/Tanggal : \_\_\_\_\_

1. Apa yang menyebabkan orang mudah percaya pada hate speech?

.....  
.....  
.....

2. Bagaimana cara merespon online hate speech?

.....  
.....  
.....

3. Apa fungsi deployment dalam game development?

.....  
.....  
.....

4. Apa keuntungan menggunakan JavaScript Library?

.....  
.....  
.....  
.....



# 16

## SUMMARY AND INTERMEDIATE 1 TEST



### Apa Yang Akan Kita Pelajari?

1. Summary meeting 9-15
2. Intermediate 1 Test



# INTERMEDIATE 2

## GAME DEVELOPMENT WITH PASHER 3





# 1

# PHASER AS A FRAMEWORK



## Apa Yang Akan Kita Pelajari?

1. Pengenalan Phaser Framework
2. Mengetahui cara menggunakan Phaser Framework
3. Mampu menjalankan code dan loading assets



# PHASER AS A FRAMEWORK



## Warm Up!



Masih ingatkah kalian apa itu **Framework**?  
**Framework** apa saja yang telah kalian ketahui?



Ada banyak sekali framework yang akan memudahkan kalian dalam membuat game. Salah satunya adalah **Phaser**!



### Vocabulary

#### Phaser

Phaser adalah framework untuk membuat game HTML 5 dalam bentuk 2D.



Kenapa harus **Phaser**?  
Menurutmu mengapa kita akan belajar **Phaser**?

Kita belajar Phaser, karena :

1. Game yang dihasilkan dapat berjalan pada **desktop** maupun **mobile browser**.
2. Mudah dalam **loading asset** (Images, Sprite Sheets, Audio, Tilemaps)
3. **Input** dari segala sisi (Multi-touch, Keyboard, Pointer, Mouse)



## What's Inside Phaser Folder?



Kalian sudah meng-**install** Phaser kan?

Sekarang yuk coba **buka folder Phaser template** yang sudah kamu download!

```
└── dist
└── node_modules
└── public
└── src
 ├── scenes
 │ ├── HelloWorldScene.js
 │ ├── index.html
 └── main.js
```



**Struktur file** pada Phaser template kamu seperti pada gambar di samping bukan?



Kira-kira apa sih yang ada di dalam folder-folder itu? Dan apa gunanya?

Isi dan kegunaan dari folder Phaser Template.

- 1. Folder dist**, folder ini berasal dari kata **directory**, berisi file production yang akan di upload ke web server.
- 2. Folder node\_modules**, folder ini berisi library yang kita butuhkan termasuk **Phaser 3**.
- 3. Folder public**, folder ini berisi **static assets** yang akan kamu gunakan dalam game seperti audio dan gambar.
- 4. Folder src**, folder ini berasal dari kata **source** yang artinya sumber, sesuai dengan artinya maka folder ini yang akan menampung sumber dari game kamu, yaitu **code**.



Meskipun terdapat banyak folder didalam Phaser template, kita hanya akan menggunakan **2** folder saja, yaitu **folder public dan src**.



## 1. Static Assets



### Vocabulary

#### Static Assets

Static assets adalah object pada game yang tidak dapat diubah oleh server. Contohnya, gambar dan audio.

Seperti penjelasan sebelumnya bahwa static assets disimpan dalam **folder public**.

Gambar disamping adalah **contoh struktur folder public** agar asset kamu tersusun dengan rapi.

```
public
 └── images
 └── my-image.png
 └── music
 └── ...
 └── sfx
 └── ...
```

Struktur folder public.

1. **Folder images**, folder ini berisi file gambar yang akan kamu gunakan dalam game, seperti sprite dan background.
2. **Folder music**, folder ini berisi file musik yang akan kamu putar selama game dimainkan.
3. **Folder sfx**, folder ini berasal dari kata **sound effect**, sesuai dengan asal katanya, folder ini berisi audio yang akan kamu gunakan untuk memberikan efek suara pada game kamu.



## 2. Where to Code?

Seperti penjelasan sebelumnya bahwa code disimpan dalam **folder src**.

Gambar disamping adalah **contoh struktur folder src** pada Phaser template.

Perhatikan penjelasan dibawah agar kamu tidak bingung apa saja yang ada di dalam file-file itu!

```
└── src
 ├── scenes
 │ └── HelloWorldScene.js
 ├── index.html
 └── main.js
```

### 1. index.html



Masih ingatkah kamu bahwa untuk menampilkan game atau aplikasi yang kamu buat dengan JavaScript membutuhkan HTML?

Nah, Phaser juga membutuhkan HTML, kita namakan **index.html**

### Syntax!

```
<html>
 <head>
 <title> Phaser Template </title>
 </head>
 <body>
 <script src="main.js"></script>
 </body>
</html>
```



Masih ingatkah kamu apa kegunaan dari tag **<title>**? Menurutmu mengapa kita memanggil file **main.js** pada tag **<script>**?



## 2. main.js

File **main.js** adalah entry point atau pintu masuk game yang kita buat. Pada **main.js** kita akan membuat **Phaser.Game** untuk mengatur beberapa objek pada game seperti lebar (**width**), tinggi (**height**), dan gravitasi (**gravity**).



### Codes

```
// import framework dari folder phaser
import Phaser from 'phaser'
import HelloWorldScene from './scenes/HelloWorld-
Scene' //import kelas HelloWorldScene

const config = {
 type: Phaser.AUTO,
 width: 480, //atur lebar
 height: 640, //atur tinggi
 physics: {
 default: 'arcade', // library Arcade Phaser
 arcade: {
 gravity: { y: 200 }, //atur gravitasi
 debug: true
 }
 },
 scene: [HelloWorldScene] //atur scene
}
export default new Phaser.Game(config)
```

**main.js** juga mengatur **scene** apa saja yang akan tampil pada game. Namun, sebelum mengatur scene, kita perlu meng-**import** scene terlebih dahulu seperti pada baris **ke-2**.



### 3. Folder scenes

Folder ini berisi **file scene** dari game yang kamu buat. Scenes menggabungkan game object dengan logika (code) yang terkait. Dimana file ini mengambil properti dari library **Phaser.Scene**.

Pada Phaser template terdapat **HelloWorldScene.js**, dimana kamu akan melihat struktur file nya seperti dibawah ini.

#### Syntax!

```
import Phaser from 'phaser'

export default class GameScene extends Phaser.Scene
{
 constructor()
 {
 super('hello-world')
 }

 preload()
 {
 // commands
 }

 create()
 {
 // commands
 }
}
```



## Pertemuan 1-Phaser as a Framework

### Load Background and Sprite

File **HelloWorldScene.js** adalah sebuah Class yang bernama HelloWorldScene. Class ini berisi 3 method, yaitu:

1. **constructor**, untuk menginisialisasikan Scene yang dibuat, sehingga dapat dipanggil kembali saat dibutuhkan.
2. **preload**, untuk memuat semua game assets yang dibutuhkan pada game.
3. **create**, untuk membangun Scene agar termuat semua game objek dan rule yang dibutuhkan.



Selain 3 method tadi, ada satu method lagi loh yang akan kamu gunakan, yaitu **update()**. Method ini akan merekam semua perubahan yang terjadi pada Scene selama game berlangsung, jadi method ini akan diulang terus menerus (**forever**).



### Coordinates System in Phaser

Sistem Koordinat adalah sistem yang menentukan posisi suatu objek dalam bentuk angka, yang terdiri dari Koordinat X dan Koordinat Y.

**Jika di Scratch,**

titik X = 0 dan Y = 0 berada di tengah-tengah layout.

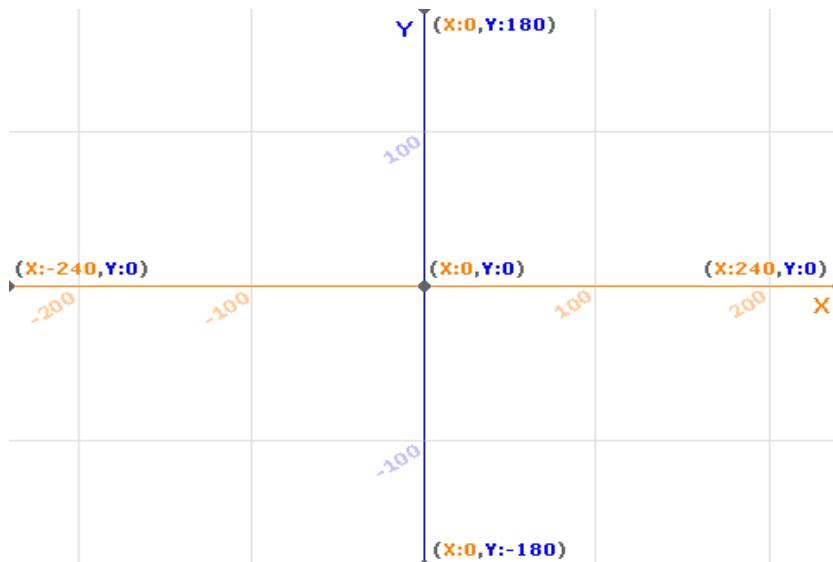
**Tapi, berbeda dengan Phaser 3.**

Phaser 3 memiliki titik x = 0 dan y = 0 di pojok kiri atas

**Coba perhatikan gambar di bawah ini ya !**



### Sistem Koordinat pada Scratch



### Sistem Koordinat pada Phaser 3

$(x,y) = (0,0)$   $(480,0)$





## Pertemuan 1-Phaser as a Framework

### Load Background and Sprite

#### Jika di Scratch,

Nilai X dan Y bisa diisi dengan nilai positif atau negatif.

Misalnya jika nilai X = -20 dan Y = 30,

Maka posisi Sprite berada di kiri atas

#### Jika di Phaser 3?

Nilai X dan Y akan **diisi dengan nilai positif saja**, karena titik X = 0 dan Y = 0 berada di pojok kiri atas.

Jika ingin menentukan posisi objek, kamu dapat mengira-ngira nilainya sesuai dengan besar layout yang dibuat.

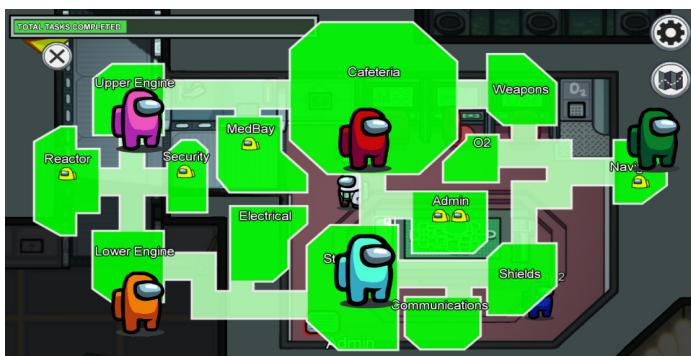
**Agar lebih paham, ayo kita coba pada latihan berikut ini !**



**Let's Practice!**

## Main Project

### Load Background and Sprite



Masih ingatkah kamu langkah pertama untuk membuat game pada Scratch?

Yup langkah pertama membuat game dalam Scratch adalah **menyiapkan background dan sprite** yang digunakan!

Sama hal nya dengan Scratch, dalam membuat game pada Phaser kita harus menyiapkan (**load**) background dan sprite terlebih dahulu!

**Yuk mulai kerjakan!**



## 1. Prepare Template

Sebelum load background dan sprite, kamu perlu menyiapkan template terlebih dahulu! Yuk ikuti langkah-langkah dibawah!

1. Duplikat folder Phaser Template, caranya **copy** folder Phaser Template dan **paste** di tempat yang sama.
2. Ganti nama (**rename**) folder hasil duplikat menjadi “**Meeting1**”.
3. Buka folder tersebut di Visual Studio Code. (**File -> Open Folder**)
4. Buat **file baru** pada folder **scenes**, caranya klik kanan pada folder scenes dan pilih New File, beri nama “**AmongUsScene.js**”.
5. Ketikkan **code** berikut pada file **AmongUsScene.js**



### Codes

```
import Phaser from 'phaser'
export default class AmongUsScene extends Phaser.
Scene
{
 constructor()
 {
 super('among-us-scene')
 }

 preload()
 {
 }

 create()
 {
 }
}
```



## 2. Download Assets

Untuk load background dan sprite, tentunya kita harus memiliki file gambar yang akan kita gunakan bukan?  
Yuk ikuti langkah-langkah dibawah!

1. Buat folder **public** di dalam folder **Meeting1**.
2. Buat folder **images** di dalam folder **public**.
3. **Download** Assets pada link berikut :  
[http://tiny.cc/meeting1\\_assets](http://tiny.cc/meeting1_assets)
4. Letakkan assets pada folder **public -> images**.



 Apakah kalian sudah melihat asset yang akan kita gunakan sebagai background dan sprite?  
Gambar apakah itu?

Yup! itu adalah gambar karakter pada game favorite kalian yaitu, **Among US**. Kita akan menggunakan **maps among us** sebagai **background** dan **karakter** (red, cyan, orange, dll) sebagai **sprite**!

## 3. Code Preload()

Pada method `preload()` kita akan meng-upload semua gambar yang dibutuhkan pada game. Perhatikan syntax berikut!

### Syntax!

```
preload()
{
 this.load.image('gameObjectKey', 'location')
}
```

### Keterangan :

gameObjectKey = kata kunci/nama dari gambar yang kamu upload  
location = lokasi gambar mu disimpan



Ketikkan code berikut untuk meng-upload Maps dan Cyan!



### Codes

```
preload()
{
 this.load.image('maps', 'images/Maps.png')
 this.load.image('playerRed', 'images/Red.png')
}
```

Sekarang coba run code yang sudah kamu buat!



Masih ingat bagaimana cara run code?

Coba buka kembali **panduan instalasi Phaser** yang telah diberikan sebelumnya, atau tanyakan kepada gurumu!

Buka folder **Meeting1** kamu pada Command Prompt dan ketik **npm run start** lalu enter!



Setelah berhasil run code, apakah kamu melihat perbedaannya dengan hasil run phaser template?

**TIDAK!** hal ini dikarenakan **AmongUsScene.js** belum di-import pada entry point yaitu **main.js**.

## 4. Code main.js

Jangan lupa **import** Scene pada **main.js** agar dapat terlihat ketika game dijalankan! **Ikuti langkah-langkah dibawah ya!**



1. Ketik code ini dibawah code **import HelloWorldScene**



### Codes

```
import AmongUsScene from './scenes/AmongUsScene'
```



2. Hapus code **import HelloWorldScene**, atau berikan comment pada code tersebut



- Load Background and Sprite

3. Ubah code `scene` menjadi seperti dibawah ini!



### Codes

```
scene: [AmongUsScene]
```

Pada langkah ini, kita menambahkan `AmongUsScene` dan menghapus `HelloWorldScene`. Artinya kita sedang memanggil AmongUsScene setelah berhasil meng-import pada langkah sebelumnya

 Apa yang terjadi pada game kamu ?

**Jangan panik** bila layar game kamu menjadi hitam.



### Do it Your Self !

Ubah **lebar** layar game menjadi **1920 (width)** dan ubah **tinggi** layar game menjadi **1080 (height)** pada `main.js`.

**Sekarang ayo kembalike AmongUsScene.js untuk membuat kode pada method `create()`!**

## 4. Code `create()`

Gambar yang kamu upload pada method `preload()` **belum dapat tampil** pada layar game. Kamu harus **menampilkannya** terlebih dahulu pada method **`create()`**. Caranya ikuti code dibawah!

### Syntax!

```
create()
{
 this.add.image(x, y, 'gameObjectKey')
}
```

### Keterangan :

x, y = Posisi koordinat **x** dan **y**

gameObjectKey = kata kunci/nama dari gambar yang kamu upload



Ketikkan code berikut untuk menampilkan 'maps' sebagai background!



### Codes

```
create()
{
 this.add.image(960, 540, 'maps')
}
```

Jangan lupa reload untuk melihat perubahannya!

Layar game kamu akan terlihat seperti gambar dibawah!



Selamat kamu telah berhasil membuat background!

**Sekarang yuk coba tampilkan sprite 'red' dan letakkan pada Cafetaria**  
Tulis code ini dibawah code maps tadi!



### Codes

```
this.add.image(1000, 400, 'playerRed')
```

### • Load Background and Sprite



Jangan lupa reload untuk melihat perubahannya!

Layar game kamu akan terlihat seperti gambar dibawah!



Selamat kamu telah berhasil membuat background dan sprite!

**Sekarang yuk coba kerjakan Challenges dibawah!**



#### Challenge!

1. Upload semua sprite yang belum terupload pada method `preload()`.
2. Letakkan `playerGreen` pada ruang Navigation.
3. Letakkan `playerPink` pada ruang Upper Engine.
4. Letakkan `playerOrange` pada ruang Lower Engine.
5. Letakkan `playerCyan` pada ruang Storage.





Tidakkah kamu merasa ukuran sprite Cyan terlalu besar?  
Menurutmu bagaimana cara mengecilkan ukuran sprite?



### Let's Explore

Untuk mengatur ukuran sprite/gambar, kamu dapat **menambahkan** code `.setScale()`. Isi berapa **skala** yang kamu inginkan di dalam tanda kurung.

**Ikuti contoh berikut untuk membuat ukuran Cyan menjadi setengah lebih kecil dari ukuran asli.**

```
this.add.image(1000, 400, 'playerRed').setScale(0.5)
```





## Pertemuan 1-Phaser as a Framework

### Check Your Understanding!



Hari/Tanggal :

1. Menurutmu, mengapa kita menggunakan Framework Phaser untuk membuat game?

.....  
.....  
.....

2. Apa itu Static Assets? Mengapa kita menggunakananya?

.....  
.....  
.....

3. Apa kegunaan method preload() ?

.....  
.....  
.....

4. Selain method constructor(), preload(), dan create(), method apa lagi yang akan kita gunakan? dan apa fungsi method tersebut?

.....  
.....  
.....





**Timedooring**  
Coding Academy

# 2

# COLLECTING STARS : MAKING GAME ENVIRONMENT



Apa Yang Akan Kita Pelajari?

1. Mampu loading sprite sheet asset
2. Mampu membuat animasi dari sprite sheet
3. Mampu membuat Static Physics Group



## COLLECTING STARS



### Warm Up!



Pernahkah kamu bermain **platformer game**?

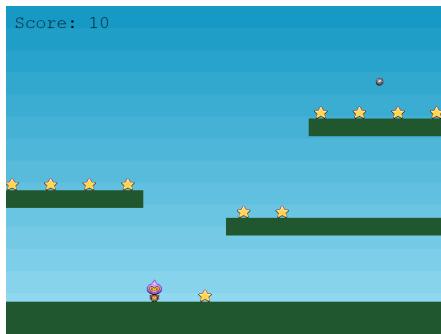
Salah satu contoh platformer game adalah **Super Mario Bros.**

Pernahkah kamu memainkannya?



Nah, kali ini kita akan membuat platformer game loh! Yuk perhatikan deskripsi Game nya!

### Collecting Stars



*Collecting Stars merupakan platformer game. Dalam game ini, terdapat satu player yang harus mengumpulkan semua bintang yang ada. Setiap satu bintang yang berhasil dikumpulkan, score akan bertambah 10. Selain itu, bom akan muncul dari atas sesuai dengan jumlah bintang yang telah dikumpulkan. Hindari bom dan kumpulkan semua bintangnya!*



Sebelum mulai mengerjakan, bisakah kamu menebak **berapa assets** yang akan kita gunakan?



## Preparation

### 1. Prepare Template

Sebelum mulai code game Collecting Stars, kamu perlu menyiapkan template terlebih dahulu! Yuk ikuti langkah-langkah dibawah!

1. Duplikat folder Phaser Template, caranya **copy** folder Phaser Template dan **paste** di tempat yang sama.
2. Ganti nama (**rename**) folder hasil duplikat menjadi “**Collecting\_Stars**”.
3. Buka folder tersebut di Visual Studio Code. (**File -> Open Folder**)
4. Buat **file baru** pada folder **scenes**, caranya klik kanan pada folder scenes dan pilih New File, beri nama “**CollectingStarsScene.js**”.
5. Ketikkan **code** berikut pada file **CollectingStarsScene.js**



## Codes

```
import Phaser from 'phaser'
export default class CollectingStarsScene extends Phaser.Scene
{
 constructor()
 {
 super('collecting-stars-scene')
 }
 preload()
 {
 }
 create()
 {
 }
}
```



## 2. Download Assets

Setelah menyiapkan template, yuk download assets yang akan digunakan pada game Collecting Stars! Ikuti langkah-langkah dibawah ya!

1. Buat folder **public** di dalam **folder Collecting\_Stars**.
2. Buat folder **images** di dalam **folder public**.
3. **Download** Assets pada link berikut :  
[http://tiny.cc/meeting2\\_assets](http://tiny.cc/meeting2_assets)
4. Letakkan assets pada **folder public -> images**.



## Loading Assets

 Masih ingatkah kamu bagaimana cara load atau **upload gambar** pada Phaser?

## 1. Code preload()

Ketikkan code berikut pada method **preload()** untuk mengupload gambar **bomb, sky, platform, dan star**.



## Codes

```
this.load.image('ground', 'images/platform.png')
this.load.image('sky', 'images/sky.png')
this.load.image('star', 'images/star.png')
this.load.image('bomb', 'images/bomb.png')
```

 Tidakkah kamu merasa ada yang **aneh** dengan code diatas? Terdapat 5 assets yang kita miliki, tetapi **mengapa hanya 4 yang di-upload?**



Hal ini dikarenakan **dude.png** bukan gambar biasa, dimana gambar tersebut berupa **sprite sheet** yang berisi beberapa frame animasi.



Apa itu Sprite sheet?

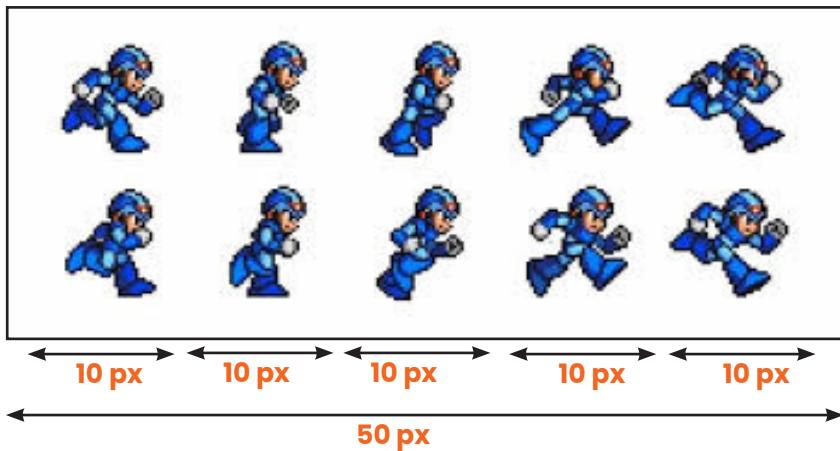


Vocabulary

### Sprite Sheet

Sprite Sheet adalah sebuah gambar yang terdiri dari beberapa gambar lainnya (Frame) dan dapat digunakan untuk membuat animasi.

Contoh sprite sheet



Pada Phaser, setiap gambar kecil tersebut disebut **Frame**. **Sprite sheet di atas memiliki 10 Frame** yang bisa digunakan untuk membuat animasi berjalan pada objek.

**Setiap Frame memiliki panjang dan lebar yang sama.**



**Jika di Scratch, Sprite sheet sama dengan Sprite Costume.**

Costume adalah animasi yang disediakan untuk sebuah Sprite

Bandingkan dengan contoh Costume di bawah ini



**Nah sekarang kita akan menggunakan Sprite Sheet dude.png pada game Collecting Star !**

 Lalu bagaimana cara load sprite sheet pada layout game?

**Perhatikan Syntax berikut untuk mengupload sprite sheet!**

### Syntax!

```
this.load.spritesheet('gameObjectName', 'location', {
 frameWidth: 10, frameHeight: 30
})
```

#### Keterangan :

- |                |                                                                 |
|----------------|-----------------------------------------------------------------|
| gameObjectName | = <b>nama</b> dari sprite sheet yang kamu upload                |
| location       | = <b>letak</b> atau lokasi dari sprite sheet mu                 |
| frameWidth     | = <b>lebar frame</b> yang ingin kamu <b>potong</b> , contoh 10  |
| frameHeight    | = <b>tinggi frame</b> yang ingin kamu <b>potong</b> , contoh 30 |



Ketikkan code berikut untuk meng-upload sprite sheet `dude.png`!



### Codes

```
this.load.spritesheet('dude', 'images/dude.png', {
 frameWidth: 32, frameHeight: 48
})
```

Yeay! semua assets telah berhasil di upload. Sekarang **yuk tampilkan assets background dan sprite-mu!**

## 2. Code create()

Seperti yang telah dijelaskan pada **pertemuan 1**, untuk menampilkan background dan sprite kita akan **menambahkan code** pada method `create()`.



Masih ingatkah kamu bagaimana caranya?

Ketikkan code berikut untuk menampilkan **sky sebagai background** dan **star sebagai sprite**!



### Codes

```
this.add.image(400, 300, 'sky')
this.add.image(400, 300, 'star')
```

Coba **run code** dan lihat apa yang terjadi!

Buka folder **Collecting\_Stars** kamu pada Command Prompt dan ketik **npm run start** lalu enter!



Apakah kamu **melihat** background **langit biru dan bintang ditengahnya?**  
Jika **tidak**, menurutmu **mengapa** hal ini bisa terjadi?

### 3. Code main.js

Background dan sprite kamu belum terlihat karena kamu belum meng-**import** dan **memanggil** kelas **CollectingStarsScene** pada **main.js**.

Yuk ikuti langkah-langkah dibawah!

1. Ketik code ini dibawah code `import HelloWorldScene`



#### Codes

```
import CollectingStarsScene from './scenes/CollectingStarsScene'
```

2. Hapus code `import HelloWorldScene`, atau berikan comment pada code tersebut

3. Ubah code `scene` menjadi seperti dibawah ini!

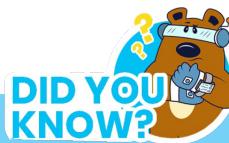


#### Codes

```
scene: [CollectingStarsScene]
```

**Jangan lupa reload untuk melihat perubahannya!**

Layar game kamu akan terlihat seperti gambar berikut ini!



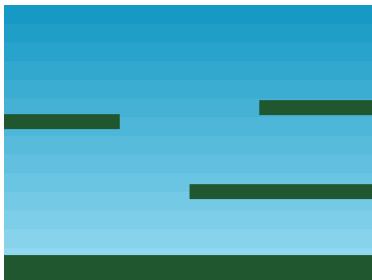
Phaser menjalankan code dari **baris paling atas hingga bawah satu persatu**. Sehingga ketika menampilkan background dan sprite, **pastikan kamu menuliskan code background terlebih dahulu** agar background tidak menutupi sprite!



## Game Environment Building

Setelah berhasil menampilkan background dan sprite, yuk kita **bangun game environment**-nya, kita akan membuat beberapa **platform** yang akan dipijak oleh player.

**Gambar dibawah adalah gambaran platform yang akan kita buat!**



Namun, sebelum membangun platform, kita perlu membuat **variable global** terlebih dahulu.



**Masih ingatkah kamu apa itu variable global? Dan tahukah kamu cara membuat variable global di**

Membuat variable global di Phaser caranya sama dengan membuat variable pada JavaScript. Ikuti instruksi dibawah ya!

```
import Phaser from 'phaser'
var platforms
export default class CollectingStarsScene
```



Ketik code berikut diantara code import dan export seperti gambar



### Codes

```
var platforms
```

## Code create()

### 1. Buat **Static Physics Group** Baru

Kita akan membuat **lebih dari satu** platform yang **dapat dipijak** oleh player, sehingga kita **tidak dapat menggunakan** `this.add.image`.

Kita akan menggunakan kelas **StaticGroup** dari **Arcade Physics** yang telah disediakan oleh Phaser.

Ketik code dibawah pada method `create()`.



### Codes

```
platforms = this.physics.add.staticGroup()
```

Nah, disinilah kita menggunakan variable global yang kita buat sebelumnya. Kita **menginisialisasikan** kelas **StaticGroup** ke dalam variable **platforms**!



Phaser menyediakan **2 kelas** untuk **grouping**, yaitu **staticGroup** dan **Group**. Kita menggunakan **staticGroup** karena **platform** yang kita buat **bersifat statis** (tidak dapat bergerak).



## 2. Buat Platform Atas

Untuk membuat platform disini caranya hampir sama dengan kita menampilkan background. **Perhatikan Syntax dibawah ini!**

### Syntax!

```
platforms.create(x, y, 'gameObjectName')
```

#### Keterangan :

platforms = **variable** yang menampung Static Physics Group (step 1)

gameObjectName = **nama** dari asset yang telah kamu upload

x, y = **letak atau posisi** berdasarkan koordinat x dan y

Ketikkan code berikut untuk membuat 3 platform atas!

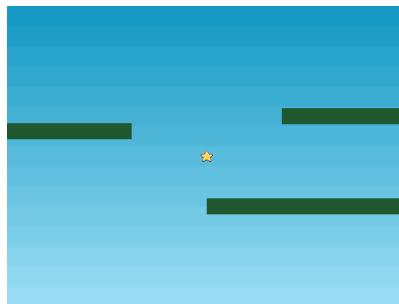


### Codes

```
platforms.create(600, 400, 'ground')
platforms.create(50, 250, 'ground')
platforms.create(750, 220, 'ground')
```

Jangan lupa reload untuk melihat perubahannya!

Layar game kamu akan terlihat seperti gambar dibawah!



## 3. Buat Platform Bawah

Untuk membuat platform bawah caranya sama dengan platform sebelumnya namun karena **ukurannya berbeda** kita akan menambahkan **2 atribut lain**.

## Pertemuan 2-Collecting Stars : Making Game Environment Create and Animate The Player



Ketikkan code berikut untuk membuat platform bawah!



### Codes

```
platforms.create(400, 568, 'ground').setScale(2).refreshBody()
```



Masih ingatkah kamu dengan atribut `setScale()`?  
Menurutmu, mengapa kita menambahkan `.w(2)` pada code diatas?

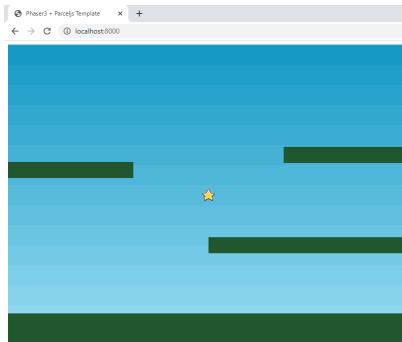


Dan untuk apakah kita memanggil `.refreshBody()`?

`.refreshBody()` berguna untuk memberitahukan kepada **Phaser (physics world)** bahwa kita telah mengubah ukuran `platforms` dari ukuran asli.

### Jangan lupa reload untuk melihat perubahannya!

Layar game kamu akan terlihat seperti gambar dibawah!



### Create and Animate The Player

Setelah berhasil membangun game environment, yuk kita buat **player**-nya!



## Code create()

### 1. Comment atau hapus code yang menampilkan 'star'

Karena kita belum menggunakan star, coba berikan comment atau hapus code `this.add.image(400, 300, 'star')` terlebih dahulu.

### 2. Buat Player

Hampir sama dengan pembuatan platform, dalam pembuatan player Kita akan menggunakan kelas **Sprite** dari **Arcade Physics** yang telah disediakan oleh Phaser.

 Menurutmu, mengapa kita menggunakan **Arcade Physics** untuk membuat player?

Kita menggunakan kelas **Sprite** dari **Arcade Physics** karena kita akan membuat **player** yang **dinamis** (dapat bergerak, meloncat, dan lain sebagainya).

**Ketikkan code berikut untuk membuat dude sebagai player!**

Sebelum mulai code, buat dulu variable global bernama **player** -->  
`var player`



### Codes

```
player = this.physics.add.sprite(100, 450, 'dude')
player.setCollideWorldBounds(true)
player.setBounce(0.2)
```

Selain membuat **dude** sebagai **player**, kita juga membuat :

1. **player collide** (berbenturan) dengan **batas game**

. `setColliderWorldBounds(true)`

2. memberikan efek sedikit memantul ketika sampai batas

. `setBounce(0.2)`



### Jangan lupa reload untuk melihat perubahannya!

Layar game kamu akan terlihat seperti gambar dibawah dan **pastikan** player tetap **berada di area game** (tidak menembus batas game).



```
player.setCollideWorldBounds(true)
player.setBounce(0.2)
```

if on edge, bounce

memiliki kegunaan yang sama dengan code pada **scratch**.



### Let's Explore

Coba berikan **comment** pada salah satu atau kedua code ini dan lihat **apa yang terjadi!**

```
player.setCollideWorldBounds(true)
player.setBounce(0.2)
```

**Jangan lupa kembalikan code seperti semula sebelum lanjut ke step berikutnya!**

### 3. Buat Animasi pada Player



Masih ingat mengapa kita upload **dude.png** sebagai **sprite sheet**?



**dude.png** terdiri dari **9 frame** animasi, dimana **frame 0-3** akan kita gunakan untuk membuat animasi player berjalan ke **kiri**, **frame 4** untuk player melihat ke **depan**, dan **frame 5-8** untuk player berjalan ke **kanan**.

Ketikkan code berikut untuk membuat animasi player!



### Codes

```
//Animasi Berjalan ke Kiri
this.anims.create({
 key: 'left', //nama animasi
 frames: this.anims.generateFrameNumbers('dude',
 { start: 0, end: 3 }), //frame yang digunakan
 frameRate: 10,
 repeat: -1
});

//Animasi Menghadap ke Depan
this.anims.create({
 key: 'turn',
 frames: [{ key: 'dude', frame: 4 }], //satu frame saja
 frameRate: 20
});

//Animasi Berjalan ke Kanan
this.anims.create({
 key: 'right',
 frames: this.anims.generateFrameNumbers('dude',
 { start: 5, end: 8 }),
 frameRate: 10,
 repeat: -1 //mengulangi animasi (loop)
});
```

## Pertemuan 2-Collecting Stars : Making Game Environment

### Create and Animate The Player



**Yeay! Selamat kamu telah berhasil membuat animasi untuk player!**

 Tetapi, **apakah ada perubahan** ketika kamu reload game kamu? Animasi player memang telah dibuat, tetapi belum dapat digunakan karena kita **belum** membuat event untuk **menggerakkan player!**

Event untuk menggerakkan player akan kita buat pada pertemuan selanjutnya. **Sekarang coba kamu kerjakan challenge berikut!**



#### Challenge!

1. Coba ganti **judul tab** game kamu menjadi "**Collecting Stars**". Caranya Buka file **index.html**, ganti judul pada tab **<title>...</title>**
2. Coba **ganti posisi platform atas** atau **tambahkan** platform lain sesukamu.

## Pertemuan 2-Collecting Stars : Making Game Environment Check Your Understanding!



Hari/Tanggal :

1. Apa kegunaan `frameWidth` dan `frameHeight` ketika kamu upload sprite sheet?

.....  
.....  
.....

2. Mengapa kita menggunakan code  
`this.physics.add.staticGroup()` untuk membuat platform?

.....  
.....  
.....

3. Mengapa kita menggunakan `.refreshBody()` ketika membuat platform bawah?

.....  
.....  
.....

4. Code apa yang kita gunakan untuk membuat player tidak keluar dari area game?

.....  
.....  
.....



# 3

# COLLECTING STARS: ADDING PHYSICS AND SCORE



**Apa Yang Akan Kita Pelajari?**

1. Mampu membuat collision
2. Membuat Keyboard Control
3. Membuat Score pada Phaser



## CONTINUE COLLECTING STARS GAME

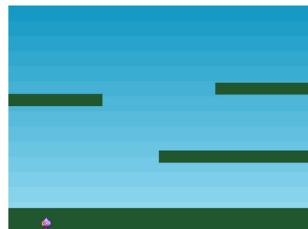


### Warm Up!

Masih ingat sampai mana kamu membuat game **Collecting Stars** pada pertemuan sebelumnya?

**Sebelum melanjutkan game Collecting Stars, yuk run terlebih dahulu game-nya!**

Apakah game kamu masih seperti gambar disamping?  
Bukankah seharusnya **player** dapat berdiri diatas **platform**?



### Adding Physics : Collider

Yuk buat player dapat berdiri diatas platform, dengan cara membuat **player** dan **platform collide** (berbenturan). Ikuti langkah-langkah dibawah ya!

**Jangan lupa, sebelum mulai code, buka dulu project kamu di Visual Studio Code!**

### Code Create()

Kita akan membuat **Collider Object** dari **Arcade Physics** yang telah disediakan oleh **Phaser**.

Ketik code dibawah pada method **create()**.



### Codes

```
this.physics.add.collider(player, platforms)
```



Yang berada di **dalam kurung** adalah **2 nama konstanta** atau variable dari **game object** yang ingin kamu buat **berbenturan (collide)**.

**Jangan lupa reload untuk melihat perubahannya!**

Pastikan **player** telah **berdiri diatas platform bawah** seperti yang terlihat pada gambar disamping!



## Keyboard Controls

 Ketika bermain game, **berapa kali** kamu menggunakan **controls**? Tentunya **berulang kali bukan**?

Nah kali ini kita akan membuat **keyboard controls** untuk menggerakkan player.

 Masih ingatkah kamu bahwa selain **constructor, preload, dan create** terdapat satu method lagi yaitu **update**?

Nah, untuk **membuat keyboard controls** kita akan menggunakan method **update()**, ini karena method **update()** akan selalu **mengulang (loop)** selama game berlangsung.

Yuk persiapkan dahulu method update yang akan kita gunakan!

Ketikkan code berikut **dibawah** method **create()**.

### Syntax!

```
update()
{
 // Keyboard Controls
}
```

Sebelum mulai code keyboard controls, buat dulu variable global bernama **cursors --> var cursors**



## 1. Code Create()

Sebelum membuat keyboard controls, kita perlu memanggil function **Keyboard Manager** yang telah dibuat oleh **Phaser** terlebih dahulu.



### Codes

```
cursors = this.input.keyboard.createCursorKeys()
```

Keyboard manager telah dipanggil dan diinisialisasikan ke dalam variable **cursors**, nah untuk memanggil **key object** mana saja yang digunakan akan ditulis pada **method update()**.

## 2. Code Update()



Masih ingatkah kamu dengan **animasi player** yang telah kamu buat pada pertemuan sebelumnya? Nah, sekarang saatnya kita menggunakan animasinya!



### Codes

```
update()
{
 // jika keyboard left arrow ditekan
 if (cursors.left.isDown) {
 // kecepatan ke samping (berjalan)
 player.setVelocityX(-160)

 player.anims.play('left', true)
 } else if (cursors.right.isDown) { //kanan
 player.setVelocityX(160)

 // memanggil animasi 'turn'
 player.anims.play('right', true)
}
```



```
 } else {
 player.setVelocityX(0) //berhenti
 // memanggil animasi 'turn'
 player.anims.play('turn')
 }

 if (cursors.up.isDown){ //atas
 // kecepatan ke atas (lompat)
 player.setVelocityY(-250)
 }
}
```

Jangan lupa reload dan coba gerakkan player dengan arrow key pada keyboard!

Pastikan player dapat berjalan dan melompat, serta animasi bergerak sesuai dengan arah!



### Collect The Stars

Sesuai dengan nama game nya, yaitu **Collecting Stars** maka kita akan mengumpulkan bintang-bintang yang berjatuhan. Tetapi, **apakah kamu sudah melihat bintang** pada game mu?

Yuk sekarang saatnya kita buat bintangnya berjatuhan!



Karena ini game **Collecting Stars**, tidak mungkin kan bintang nya hanya 1?

Jadi, kita akan menggunakan kelas **Group** dari **Arcade Physics** yang telah disediakan oleh **Phaser** untuk menggandakan bintangnya.



Masih ingatkah kamu, kita pernah menggunakan kelas **StaticGroup** untuk menggandakan platform? **Menurutmu mengapa kita tidak menggunakan StaticGroup lagi pada kasus ini?**

## Pertemuan 3-Collecting Stars :Adding Physics and Score

### Collect The Stars



Alasan kita menggunakan kelas **Group** karena bintang yang kita buat bersifat **dinamis** (dapat bergerak dan memantul), sedangkan **apabila** kita menggunakan **StaticGroup** maka bintang akan bersifat **statis** (tidak dapat bergerak).

Sebelum mulai code bintang, buat dulu variable global bernama **stars** (`var stars!`)!

### 1. Code Create()

Ketikkan code berikut untuk membuat menjatuhkan bintang!



#### Codes

```
stars = this.physics.add.group({
 key: 'star', //game object name (from preload)
 repeat: 11, // mengulangi 11 kali
 // posisi bintang pertama, dan jarak dengan
 // bintang selanjutnya
 setXY: { x: 12, y: 0, stepX: 70 }
});

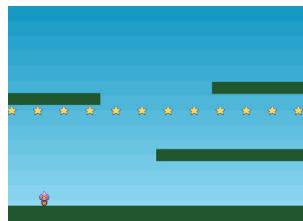
// loop semua bintang
stars.children.iterate(function (child) {
 // memberikan efek memantul dengan nilai
 // antara 0.4 - 0.8
 child.setBounceY(Phaser.Math.FloatBetween(0.4, 0.8));
});
```

Jangan lupa reload dan lihat apa yang terjadi!



Apakah kamu melihat sesuatu yang aneh disini?

Bagaimana kita dapat mengumpulkan bintang apabila bintangnya jatuh sampai bawah?





Ingatkah kamu kita pernah membuat player dan platform menjadi **collide**? Yuk coba buat **stars** dan **platform collide**, agar player dapat mengumpulkan stars nanti!

Ketikkan code berikut untuk membuat stars dan platform collide!



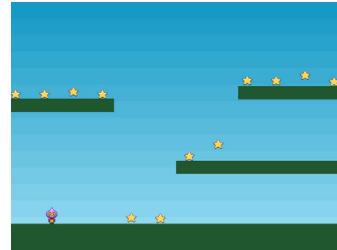
### Codes

```
this.physics.add.collider(stars, platforms)
```

Jangan lupa reload dan lihat perubahannya!

Pastikan **stars** dan **platform** telah **collide** (berbenturan), serta **bintang** telihat **memantul** ketika **menyentuh platforms**!

Sekarang, coba gerakkan player hingga menyentuh bintang!



Apakah player dapat mengambil bintangnya?

Belum! Yuk kita buat player overlaps dengan bintang!

Ketikkan code berikut untuk membuat player dan bintang overlap!



### Codes

```
this.physics.add.overlap(
 player, // membuat overlap antara player
 stars, // dan stars
 this.collectStar, // memanggil method collectStar
 null, // proses callback yang tidak dibutuhkan
 this // memastikan overlap pada scene ini
)
```

Code diatas membuat **player** dan **stars overlap**, tetapi **belum** memberikan **efek** yang terjadi. Oleh karena itu, pada code ini kita **memanggil method collectStar** untuk memberikan efek yang terjadi seperti salah satu game object menghilang atau score bertambah.



### Yuk kita buat method collectStar!

 Masih ingatkan kamu bagaimana cara membuat **method baru?** Caranya **hampir sama** seperti kita membuat method **update()**.

Ketikkan code berikut **dibawah** method **create()** dan sebelum **update()**.



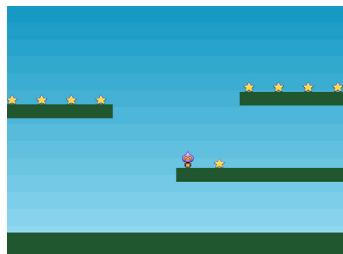
### Codes

```
// method collectStar dengan parameter player dan star
collectStar(player, star)
{
 // menghilangkan fisik star/bintang
 star.disableBody(true, true)
}
```

**Jangan lupa reload dan coba gerakkan player hingga menabrak bintang!**

Pastikan **bintang** yang **menghilang!** Bukan **player!**

Yeay! selamat game kamu telah berhasil membuat player mengumpulkan star!



 Kalau hanya mengumpulkan bintang tetapi tidak dapat score, apa kamu masih semangat bermain game nya?

**Yuk tambahkan score agar player semakin semangat bermain game-mu!**



### Scoring

 Masih ingatkah kamu cara membuat score di **Scratch**??? Apa **langkah pertama** yang kamu lakukan untuk membuat score???

Langkah pertama untuk membuat score adalah membuat **variable terlebih dahulu!** Yuk buat **variable global** dengan nama **score**, set **nilai** variable **score** menjadi **0 --> var score = 0**. Serta variable global ber-nama **scoreText --> var scoreText**.



Tahukah kamu **perbedaan** dari variable **score** dan **scoreText**?  
Yup! **score** untuk **menyimpan nilai**-nya, sedangkan **scoreText** untuk **menampilkan text**-nya.

## 1. Code Create()

Untuk menambahkan text pada game, kita dapat menggunakan syntax dibawah!

### Syntax!

```
this.add.text(x, y, 'text', {
 fontSize: '20px', fill: '#000'
});
```

### Keterangan:

x, y	= letak atau posisi berdasarkan koordinat x dan y
text	= text yang akan ditampilkan pada layar game
fontSize	= ukuran font, dalam satuan pixel (px) seperti contoh
fill	= warna text, dapat diisi dengan kode warna hexadecimal seperti contoh, atau dengan kode warna basic seperti ' <b>red</b> ', ' <b>black</b> ', ' <b>yellow</b> ', ' <b>green</b> ', dll.

Sekarang yuk kita buat tampilan atau style dari **scoreText**. Ikuti code di bawah ya!



### Codes

```
scoreText = this.add.text(16, 16, 'score : 0', {
 fontSize: '32px', fill: 'yellow'
});
```

Code diatas membuat text "**score : 0**" dan **dinisialisai**kan ke dalam variable **scoreText**. Ini akan **memudahkan** untuk **mengganti nilai score** ketika pemain berhasil mengumpulkan bintang!

**Coba reload** dan pastikan kamu melihat tulisan "**Score : 0**" pada bagian ujung kiri atas layar game kamu, seperti gambar disamping!

Score : 0



## 2. Code collectStar()

Tampilan score sudah kita buat, sekarang saatnya kita menambahkan nilai score ketika player berhasil mengumpulkan bintang!

Untuk mengubah nilai score, tuliskan code dibawah pada method `collectStar()`.



### Codes

```
score += 10 // 1 bintang bernilai 10
// menampilkan text dengan nilai score terbaru
scoreText.setText('Score : ' + score)
```

#### Reload dan coba mainkan game-nya!

Pastikan score bertambah ketika player overlap dengan bintang!

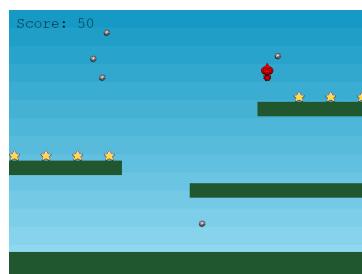
Score : 120

 Sudahkah kamu memainkan game Collecting Stars yang kamu buat? Tidakkah kamu **merasa ada yang kurang** dengan game mu? Kurang sesuatu yang lebih **menantang** bukan?

Agar game kamu lebih menarik, yuk **tambahkan bom** yang akan muncul **setiap player mengumpulkan bintang**, sehingga **semakin banyak bintang** yang dikumpulkan **semakin banyak** pula **bom** yang berkelebaran dan permainan akan menjadi semakin seru!!!



### Bouncing Bombs



Perhatikan algoritma **bouncing bombs** dibawah agar kamu tidak bingung nanti!

1. Jumlah bom yang akan muncul sesuai dengan bintang yang dikumpulkan (**lebih dari 1**)
2. Bom muncul dari **atas ke bawah** dengan posisi **x** yang diambil **secara acak** (random)



3. Bom akan **memantul** ketika menyentuh **tepi area game** (if on edge, bounce)
4. Percepatan x (**velocityX**) bom diambil secara acak (**random**), dan percepatan y (**velocityY**) bernilai **20**.
5. Buat bom **collide** (berbenturan) dengan **platform**.
6. Buat bom **collide** (berbenturan) dengan **player**, maka akan terjadi :
  - a. semua **fisik** game akan **berhenti** (pause)
  - b. warna **player** menjadi **merah**
  - c. **player** menghadap ke depan sesuai dengan **animasi 'turn'** yang telah dibuat
  - d. set **gameOver = true**

Sekarang yuk kita code bouncing bombs sesuai dengan algoritma diatas!

Sebelum mulai code bouncing bombs, buat dulu variable global bernama **bombs** dan **gameOver**. Inisialisasikan gameOver dengan nilai **false**.

## 1. Code Create()



### Codes

```
// algoritma no 1
bombs = this.physics.add.group()
// algoritma no 5
this.physics.add.collider(bombs, platforms)
// algoritma no 6 saja
this.physics.add.collider(
 player,
 bombs,
 this.hitBomb, // buat method baru
 null,
 this
);
```



## 2. Code collectStar()



### Codes

```
// algoritma no 2, posisi x diambil acak
var x = (player.x < 400) ?
 Phaser.Math.Between(400, 800) :
 Phaser.Math.Between(0, 400)

// algoritma no 2
var bomb = bombs.create(x, 0, 'bomb')
bomb.setBounce(1) // algoritma no 3, memantul
// algoritma no 3, ketika menyentuh tepi game
bomb.setCollideWorldBounds(true)
// algoritma no 4
bomb.setVelocity(
 Phaser.Math.Between(-200, 200), 20
)
```

## 3. Code hitBomb()



### Codes

```
hitBomb(player, bomb)
{
 this.physics.pause() // algoritma no 6a
 player.setTint(0xff0000) // algoritma no 6b
 player.anims.play('turn') // algoritma n 6c
 gameOver = true // algoritma no 6d
}
```

**Reload dan coba mainkan game-nya!**

Pastikan **bom** telah berjalan **sesuai** dengan **algoritma**!



Apakah kamu berhasil mengumpulkan semua bintangnya?  
Apa yang terjadi ketika semua bintang berhasil dikumpulkan?  
Apakah bintang muncul kembali atau tidak?

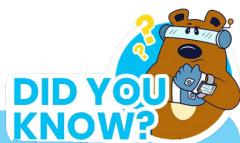
Sayangnya, bintang tidak muncul kembali saat semuanya berhasil di kumpulkan. **Kerjakan Challenge dibawah untuk membuat game mu lebih seru lagi dengan bintang yang terus bermunculan dan bom semakin banyak!**



### Do it Your Self !

Buat bintang muncul lagi ketika semuanya telah dikumpulkan dengan cara ketik code berikut pada method **collectStar()** dan ikuti instruksi yang di comment.

```
if (stars.countActive(true) === 0)
{
 stars.children.iterate(function(child){
 child.enableBody(
 status reset, //-> berisi true/false
 x, //-> posisi x
 y, //-> posisi y
 status enableGameObject, -> berisi true/false
 status showGameObject -> berisi true/false
)
 })
}
```



**Phaser.Math.FloatBetween(0.2 - 0.8)** dan  
**Phaser.Math.Between(-200, 200)**

memiliki kegunaan yang **sama** dengan code pada **scratch**.

pick random  to

## Pertemuan 3-Collecting Stars :Adding Physics and Score Check Your Understanding!



Hari/Tanggal : \_\_\_\_\_

1. Dimana kah kamu membuat code keyboard controls? Dan mengapa?

.....  
.....  
.....

2. Mengapa kita memilih menggunakan kelas Group daripada StaticGroup untuk menggandakan bintang?

.....  
.....  
.....

3. Coba jelaskan apa kegunaan dari code dibawah  
`Phaser.Math.FloatBetween(1.2 - 2.5)`

.....  
.....  
.....

4. Code apa yang kita gunakan untuk membuat collision, baik overlap atau collide?

.....  
.....  
.....



**Timedooring**  
**Coding Academy**



# 4

# BUNNY JUMP : FOLLOW THE PLAYER



## Apa Yang Akan Kita Pelajari?

1. Mampu membuat multiple platform.
2. Mampu menggunakan checkCollision dan Camera
3. Mampu mengganti animasi dengan setTexture.



Pertemuan

4

## BUNNY JUMP



### Warm Up!



Masih ingatkah kamu dengan **platformer game**?

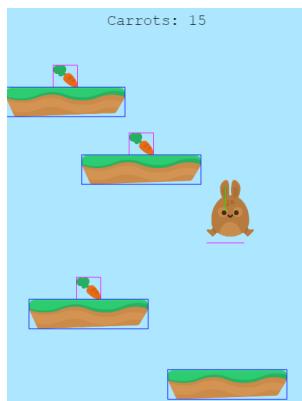
Selain Super Mario Bros ada juga contoh platformer game lain loh, yaitu **Sonic Jump**.

Pernahkah kamu memainkannya?



Nah, kita akan membuat platformer game lagi, tetapi kali ini berbeda dengan Collecting Stars yang telah kita buat! Yuk perhatikan desripsi Game nya!

## Bunny Jump



Bunny Jump termasuk dalam platformer game. Dalam game ini, terdapat satu kelinci (Bunny) yang kelaparan. Makanan Bunny (Carrot) terdapat diatas platform, Bunny harus melompati setiap platform untuk mengumpulkan carrot. Bunny akan melompat ketika mengenai platform dan mengumpulkan carrot yang ada pada platform. Jumlah carrot yang berhasil dikumpulkan Bunny akan tertulis diatas game. Apabila Bunny terjatuh dari platform maka game akan berakhir. Yuk bantu Bunny mengumpulkan wortel sebanyak-banyaknya!



Sebelum mulai code, yuk coba tebak **berapa assets** yang akan kita gunakan?



## Preparation

### 1. Prepare Template

Yuk ingat kembali cara **menyiapkan template** sebelum membuat game! Kamu bisa membuka **Pertemuan 1 atau 2** jika lupa, dan tanyakan Gurumu jika kesulitan.

- Beri nama folder game kamu "**Bunny\_Jump**".
- Buat **file baru** pada folder **scenes**, beri nama "**BunnyJump.js**".
- Ketikkan **code** berikut pada file **BunnyJump.js**



### Codes

```
import Phaser from 'phaser'
export default class BunnyJumpScene extends Phaser.Scene
{
 constructor()
 {
 super('bunny-jump-scene')
 }
 preload()
 {
 }
 create()
 {
 }
 update()
 {
 }
}
```

## Pertemuan 4-Bunny Jump : Follow The Player

### Preparation



Selanjutnya,

Pada file **index.html**, ubah judul *Phaser3 + Parceljs Template* menjadi **Collecting Stars**. Nantinya itu akan mengubah judul tab pada browser



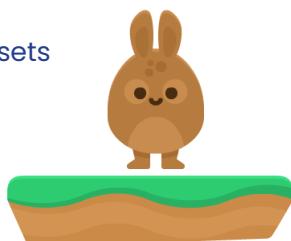
## 2. Download Assets

Setelah menyiapkan template, yuk download assets yang akan

Link download assets : [http://tiny.cc/meeting4\\_assets](http://tiny.cc/meeting4_assets)

Jangan lupa buat folder **public** yang berisi folder **images** dan **sfx**!

- Letakkan asset gambar pada **folder public -> images**.
- Letakkan asset sound effect pada **folder public -> sfx**.



## 3. Configuration

- Coba perhatikan kembali **gambar** pada **penjelasan Bunny Jump!** Apakah kamu melihat sesuatu yang **berbeda** dari **game sebelumnya**?



**Perbedaan** game Bunny Jump dengan game sebelumnya terletak pada **ukuran area game**, dimana game sebelumnya berbentuk persegi dengan ukuran 800 x 600 (sesuai dengan original phaser template). Sedangkan game **Bunny Jump** yang akan kita buat lebih potrait. Namun, selain mengatur ukuran game, kita juga akan meng-**import** scene utama game agar kamu tidak perlu kembali ke **main.js** di tengah-tengah coding.

**Yuk ikuti langkah-langkah dibawah!**

1. Buka file **main.js** pada **Visual Studio Code**.

2. Ganti lebar game (**width**) menjadi **480** dan tinggi game tinggi (**height**) menjadi **640**. Code mu akan terlihat seperti dibawah!



### Codes

```
width : 480,
height : 640,
```

3. Import scene utama pada game ini yaitu **BunnyJumpScene**. Hapus atau berikan comment pada `import HelloWorldScene` dan **tulis code dibawah**.



### Codes

```
import BunnyJumpScene from './scenes/BunnyJumpScene'
```

4. Panggil **BunnyJumpScene** pada `scene` dan hapus `HelloWorldScene`. **Code kamu akan menjadi seperti dibawah ini!**



### Codes

```
scene: [BunnyJumpScene]
```

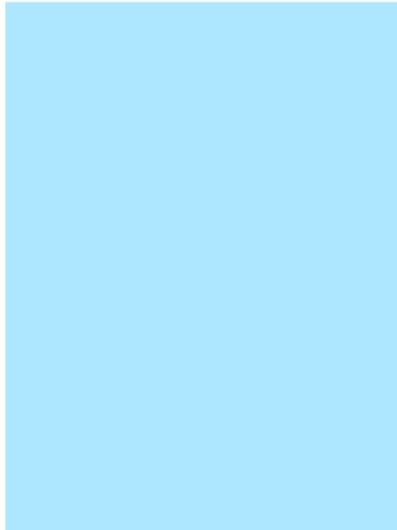
## Pertemuan 4-Bunny Jump : Follow The Player

### Preparation



 Masih ingat **cara run** game kan?  
Buka **folder Bunny\_Jump** pada command prompt dan jalankan dengan **npm run start!**

**Coba run game kamu dan pastikan layar game terlihat potrait seperti ini !**



### Creating Background and Platform

 Masih ingat kah kamu cara membuat background dan menampilkan sprite atau game object?



#### 1. Upload Image Assets

Sama seperti game yang telah kita buat, sebelum menampilkan background atau sprite, kita harus **upload** terlebih dahulu **asset yang dibutuhkan!**

Buat kode ini pada method **preload()**.



## Codes

```
this.load.image('background', 'images/bg_layer1.png')
this.load.image('platform', 'images/ground_grass.png')
```

Nah sekarang kamu **HARUS** kerjakan Challenge ini untuk load gambar yang lain ya!



### Do it Your Self !

Upload asset gambar lain yang belum ter-upload, berikan nama game object sesuai ketentuan berikut agar kamu tidak bingung nantinya!

#### FILE NAME

carrot.png

→

#### GAME OBJECT NAME

'carrot'

bunny\_jump.png

→

'bunny\_jump'



## 2. Tampilkan Background dan Platform



Masih ingat caranya menampilkan gambar?  
Ikuti contoh code dibawah dan ingat selalu ya! Code pada method `create()`.



## Codes

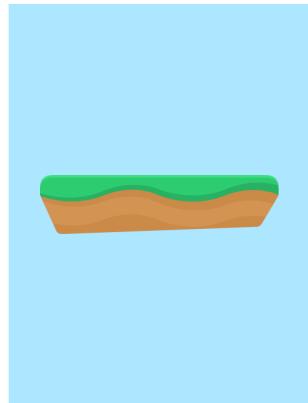
```
this.add.image(240, 320, 'background')
this.add.image(240, 320, 'platform')
```

## Pertemuan 4-Bunny Jump : Follow The Player Tampilkan Background dan Platform



Coba **reload** dan kamu akan melihat **background** berwarna **biru** dan **satu platform** berukuran **besar ditengah** layar kamu, seperti pada gambar disamping.

 Judul game kita adalah Bunny Jump. Menurutmu, apakah mungkin platform sebagai media lompatnya hanya ada satu?



**Yuk buat platform-nya menjadi lebih banyak!**

### 3. Gandakan Platform

 Masih ingat dengan konsep platform yang kita gunakan pada game Collecting Stars?

Sama seperti platform pada game Collecting Stars, kita akan membuat **Static Physics Group** lagi. Ini karena **platform tidak dapat bergerak**, dan Bunny lah yang akan bergerak untuk memijak platform selanjutnya.

Sebelum membuat Static Physics Group, **buat variable global** bernama **platforms** terlebih dahulu! (`var platforms`)

Ketikkan **code** berikut pada method `create()`.



#### Codes

```
this.platforms = this.physics.add.staticGroup()
```

Code diatas berguna untuk **menginisialisaikan Static Physics Group** ke dalam **variable platforms**!



Sekarang saatnya kita menggandakan platformnya, kita membutuhkan **5 platform** dalam satu area game.



Untuk **menggandakan** platform kita akan menggunakan **kONSEP for loop**. Masih ingatkah kamu dengan **for loop**?

Lanjutkan code berikut di dalam method **create()**.



### Codes

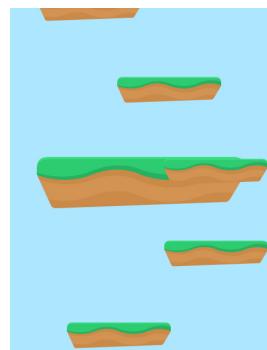
```
for (let i = 0; i < 5; i++) { // mengulangi 5 kali
 // x bernilai random dari 80-400
 const x = Phaser.Math.Between(80, 400)
 const y = 150 * i // platform akan berjarak 150 px
 // membuat platform
 const platformChild = this.platforms.create(x, y, 'platform')
 platformChild.setScale(0.5) //mengecilkan platform
 platformChild.refreshBody() //merefresh platform.
 const body = platformChild.body
 body.updateFromGameObject()
}
```

Coba **reload** dan kamu akan melihat **5 platform kecil dan 1 platform besar seperti pada gambar disamping!**



Apakah kamu melihat masalah disini?  
Yup! masalahnya ada pada platform besar. Kita tidak menggunakan platform besar.

Hapus atau berikan comment pada code  
`this.add.image(240, 320, 'platform')`

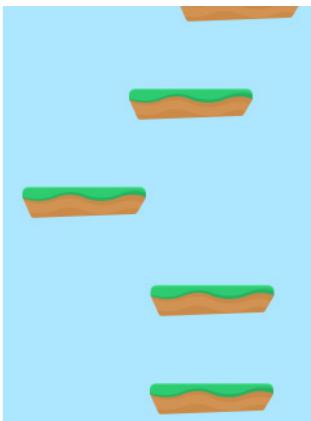


## Pertemuan 4-Bunny Jump : Follow The Player

### Membuat Player



Kemudian reload, dan pastikan platform besar telah hilang!  
Kamu bisa me-reload untuk melihat platform berubah posisi!



### Create The Player

Platform sudah kita buat, saatnya kita membuat player!



#### 1. Membuat Player

 Masih ingat bagimana caranya membuat player pada game sebelumnya?

Masih sama dengan game Collecting Stars, untuk membuat player kita akan menggunakan kelas **Sprite** dari **Arcade Physics**, karena seperti judulnya (**Bunny Jump**), player akan dapat **bergerak** dan **melompat**!

Jangan lupa **buat variable global** bernama **player** terlebih dahulu!  
`(var player)`

**Ketikkan code dibawah pada method `create()` ya!**



#### Codes

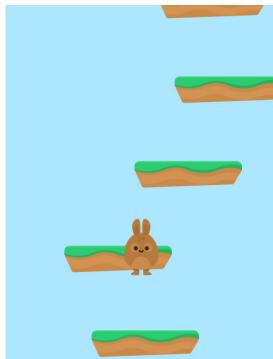
```
this.player = this.physics.add.sprite(240, 320, 'bunny_stand')
.setScale(0.5)
```



Coba **reload** dan kamu akan melihat **Bunny yang muncul di tengah layar namun jatuh kebawah, seperti gambar disamping!**

Menurutmu mengapa Bunny jatuh kebawah dan tidak dapat memijak platform?

Untuk menguji pemahamanmu, coba jelaskan code sebelumnya dan tuliskan pada kolom dibawah!



## 2. Membuat Collision

Masih ingat apa kegunaan collision?  
Bagaimana cara membuat collision?

Sesuai dengan artinya, **collision membuat 2 game object terlihat bertabrakan**. Pada game ini kita akan membuat collision antara Bunny dan Platform, sehingga **Bunny dapat berdiri diatas platform!**

Caranya gampang banget! Ikuti pentunjuk dibawah ya! Jika kamu lupa bisa buka **Pertemuan 3** atau tanyakan pada Gurumu!

- Ketik code `this.physics.add.collider()` pada method **create**.
- Tuliskan **parameter** pada code tersebut dengan 2 game object yang akan kita buat collide, yaitu **this.player** dan **this.platforms**
- Simpan dan Reload**, lihat apakah Bunny dapat berdiri diatas platform?



Mudah bukan untuk membuat collision?  
**Ingin-ingat 3 langkah diatas ya!**

## Buat Bunny Melompat



### 3. Buat Bunny Melompat

Sekarang saatnya kita buat Bunny melompat ketika mengenai platform!

Karena melompat akan terjadi **sepanjang game dimainkan**, maka kita akan membuat code pada method **update()**.

Ketik code berikut untuk membuat Bunny melompat!



#### Codes

```
update()
{
 // variable lokal untuk memastikan player menyentuh bawah
 const touchingDown = this.player.body.touching.down

 //kondisi jika player menyentuh bawah
 if(touchingDown)
 {
 //maka player akan meloncat dengan percepatan -300
 this.player.setVelocityY(-300) // -300 karena keatas
 //dan berubah animasi menjadi melompat
 this.player.setTexture('bunny_jump')
 }
}
```



Coba **reload** dan pastikan **Bunny** telah **melompat** ketika **mengenai platform**!



**Apakah kamu menemukan masalah lagi disini?**

1. Bunny tidak dapat kembali ke animasi semula setelah melompat
2. Bunny tidak dapat melompat lebih tinggi karena terhalang platform lain



**Yuk kita selesaikan masalah pertama dulu!**

Ketik code dibawah **setalah** code tadi! (Masih dalam method **update()**)



## Codes

```
//mencari percepatan player
const vy = this.player.body.velocity.y
//jika percepatan lebih dari 0 dan animasi player bukan
//stand/berdiri
if (vy > 0 && this.player.texture.key !== 'bunny_stand')
{
 // buat animasi player berdiri/stand
 this.player.setTexture('bunny_stand')
}
```

Coba **reload** dan pastikan **Bunny terlihat berdiri** setelah melompat, seperti yang terlihat pada gambar disamping!

**Yeay! Selamat masalah pertama telah berhasil kamu selesaikan!**

Untuk menyelesaikan masalah kedua, yuk perhatikan poin selanjutnya!



## 4. Check Collision

Masalah kedua disebabkan oleh **collision** yang kita buat **menyeluruh** antara **Bunny** dan **Platform**. Sedangkan kita ingin agar **Bunny** dan **Platform** hanya **collide** pada bagian **ketika Bunny memijak platform**.

Phaser telah menyediakan properti **checkCollision** dari kelas **Phaser.Physics.Arcade.Body** yang dapat kita gunakan untuk kasus ini !

## Pertemuan 4-Bunny Jump : Follow The Player

### Ikuti Bunny



Yuk, ketik code berikut pada method `create()`!



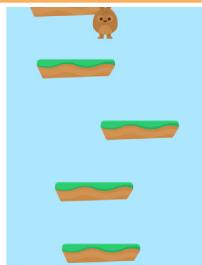
#### Codes

```
// mematikan collision player pada bagian atas, kiri, dan
// kanan
this.player.body.checkCollision.up = false
this.player.body.checkCollision.left = false
this.player.body.checkCollision.right = false
```

Coba **reload**, pastikan **Bunny** dapat **melompat** melwati platform atasnya dan **melompat lagi**!



Tetapi, apakah kamu melihat masalah disini?  
Yup, Bunny akan menghilang diatas area game.



#### 5. Ikuti Bunny

Agar Bunny tidak menghilang diatas area game, yuk kita ikuti saja kemana Bunny pergi!

Untuk mengikuti Bunny, kita akan menggunakan **Camera** yang telah disediakan oleh Phaser.

Yuk, ketik code berikut pada method `create()`!



#### Codes

```
// membuat camera mengikuti player
this.cameras.main.startFollow(this.player)
```



Coba **reload**, pastikan layar telah bergerak **keatas** mengikuti **Bunny** dan sampai ke **ujung** platform seperti gambar disamping!



Nah sampai sini, menurutmu kira-kira apa masalah yang harus kita selesaikan di pertemuan selanjutnya?



Hari/Tanggal : \_\_\_\_\_

1. Konsep apa yang kita gunakan untuk menggandakan platform?  
Dan mengapa?

.....  
.....  
.....

2. Bagaimana cara membuat suatu game object collide / bertabrakan?

.....  
.....  
.....

3. Code apa yang kita gunakan untuk membuat animasi Bunny seolah-olah seperti melompat?

.....  
.....  
.....

4. Menurutmu, apa kegunaan checkCollision?

.....  
.....  
.....

5. Bagaimana cara menggunakan camera untuk mengikuti salah satu game object?

.....  
.....  
.....



# 5

# BUNNY JUMP : SCROLLING PLATFORM



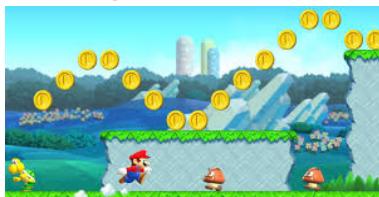
## Apa Yang Akan Kita Pelajari?

1. Membuat Animasi Player Bergerak ke Kanan dan ke Kiri
2. Membuat Scrolling Platform

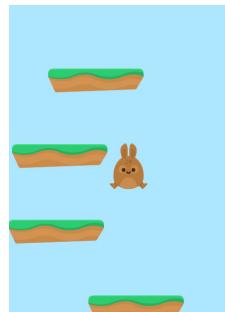


## BUNNY JUMP

Kali ini kita akan melanjutkan game Bunny Jump dengan membuat **Player dapat bergerak ke kanan dan ke kiri** dan membuat **Scrolling Platform**.



Contoh game yang menggunakan Scrolling Platform adalah **Super Mario**.



**Bunny Jump**



### Moving Left and Right



Masih ingat cara menggerakkan player di game Collecting Star? Variabel apa yang harus ditambahkan terlebih dahulu?

- Deklarasikan variabel cursors

```
var cursors
```

- Inisialisasi cursors pada method `create()`

```
this.cursors = this.input.keyboard.createCursorKeys()
```



### 1. Cursors Left and Right

Susunlah kode berikut ini pada method `update()`



#### Codes

```
if (this.cursors.left.isDown && !touchingDown) {
 this.player.setVelocityX(-200)
} else if (this.cursors.right.isDown && !touchingDown) {
 this.player.setVelocityX(200)
} else {
 this.player.setVelocityX(0)
}
```



Coba **reload** dan lihat hasilnya!

Pastikan player **sudah dapat digerakkan ke kanan dan kiri**, serta **terjatuh** ketika **tidak menekan keyboard** apapun.



## Scrolling Platform

Scrolling Platform artinya membuat platform dapat ditampilkan terus menerus mengikuti arah player.

Tapi, sebelum bisa membuat scrolling platform kita harus tahu konsep-konsep ini dulu ya!

1. Konsep Parent dan Children
2. Konsep Iterasi
3. Konsep ScrollY



### 1. Parent dan Children

Pada game ini, kita akan menemukan konsep Parent-Children yang terdapat pada JavaScript.



Apa itu Parent-Children?



#### Vocabulary

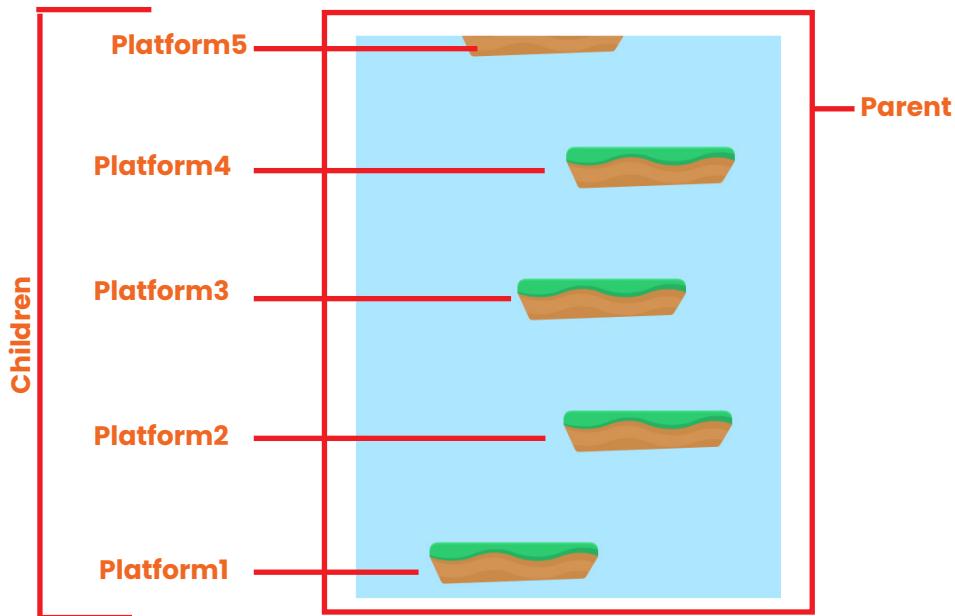
**Parent** adalah kelas yang mendefinisikan properti dan metode untuk dirinya sendiri dan juga untuk kelas Child nya.

**Child** adalah kelas yang menggunakan properti dan metode yang berasal dari kelas Parent nya



## Scrolling Platform

Dalam game Bunny Jump, **Platforms sebagai objek grup adalah kelas Parent** yang memiliki **children** yaitu **Platform1, Platform2, Platform3, Platform4, dan Platform5**



Platform 1 = child

Platform 2 = child

Platform 3 = child

Platform 4 = child

Platform 5 = child

Children

Jika **Platforms** ditampilkan dalam bentuk **Array** akan menjadi seperti ini:

```
platforms = [Platform1, Platform2, Platform3, Platform4, Platform5]
```



Selanjutnya kita akan **menggunakan konsep Children** tersebut untuk menampilkan Platform **secara berulang (looping)**. Nah untuk itu yuk kita belajar **konsep Iterasi dalam looping** terlebih dahulu!



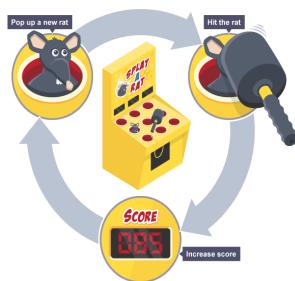
## 2. Iterasi Platforms



### Vocabulary

**Iterasi** adalah metode untuk menampilkan Array dalam bentuk perulangan.

#### Contoh.



Bagaimana konsep Iterasi pada game di samping?

Seperti yang terlihat pada gambar, ada 9 lubang yang dapat memunculkan Tikus. Itu berarti jumlah Tikus = jumlah Lubang.

**GrupTikus** = Parent  
**Tikus1** = child  
**Tikus2** = child  
**Tikus3** = child  
**Tikus4** = child  
**Tikus5** = child  
**Tikus6** = child  
**Tikus7** = child  
**Tikus8** = child  
**Tikus9** = child

Iterasi digunakan untuk **menampilkan Tikus1 sampai Tikus9 bergantian secara terus menerus**

Iterasi juga bisa memiliki **kondisi** di dalamnya.

Misalnya pada game tikus berisi sebuah kondisi:

**"Jika tikus terkena pukulan, maka score bertambah"**



## Pertemuan 5-Bunny Jump : Scrolling Platform

### Scrolling Platform

Sekarang ayo kita gunakan konsep Parent-Children dan Iterasi pada Platforms di Game Bunny Jump!

**Yuk, ketik code berikut pada method update()!**



#### Codes

```
this.platforms.children.iterate(child => {
 const platformChild = child
})
```

Kode diatas artinya melakukan Iterasi terhadap setiap Child yang terdapat pada Platforms.

```
const platformChild = child
```

Kode ini ditulis untuk mengubah variabel child menjadi platformChild.

Fungsinya hanya untuk memudahkan kita mengingat nama variabel. Artinya kamu juga bisa tetap menggunakan variabel child saja



**DID YOU KNOW?**

Selanjutnya, kita juga akan menambahkan sebuah **kondisi di dalam Iterasi Platform**. Tapi sebelum itu kita perlu belajar tentang **konsep scrollY** dulu ya!



### 3. ScrollY



#### Vocabulary

**scrollY** adalah posisi vertical kamera saat di scroll.

Nilai scrollY bisa berubah tergantung perpindahan yang dilakukan kamera.



Nah selanjutnya tambahkan kondisi di dalam perulangan agar :

**“Jika posisi Y platform lebih besar atau sama dengan scrollY ditambah 700 pixels,  
Maka pindahkan platform ke posisi Y acak antara 50-100 pixels diatas kamera”**

Lanjutkan kode ini di dalam Iterasi ya!



## Codes

```
const scrollY = this.cameras.main.scrollY
if (platformChild.y >= scrollY + 700) {
 platformChild.y = scrollY - Phaser.Math.Between(50, 100)
 platformChild.body.updateFromGameObject()
}
```

Coba reload dan mainkan game mu!



## Challenge!

Coba kamu ganti angka 700 dengan angka-angka ini dan lihat perbedaannya!

- 400
- 600
- 900



Apakah ada yang kurang dari Scroling Platform yang kamu buat?

Selain Platform, kita juga perlu membuat **background** tetap terlihat meskipun player terus bergerak ke atas.

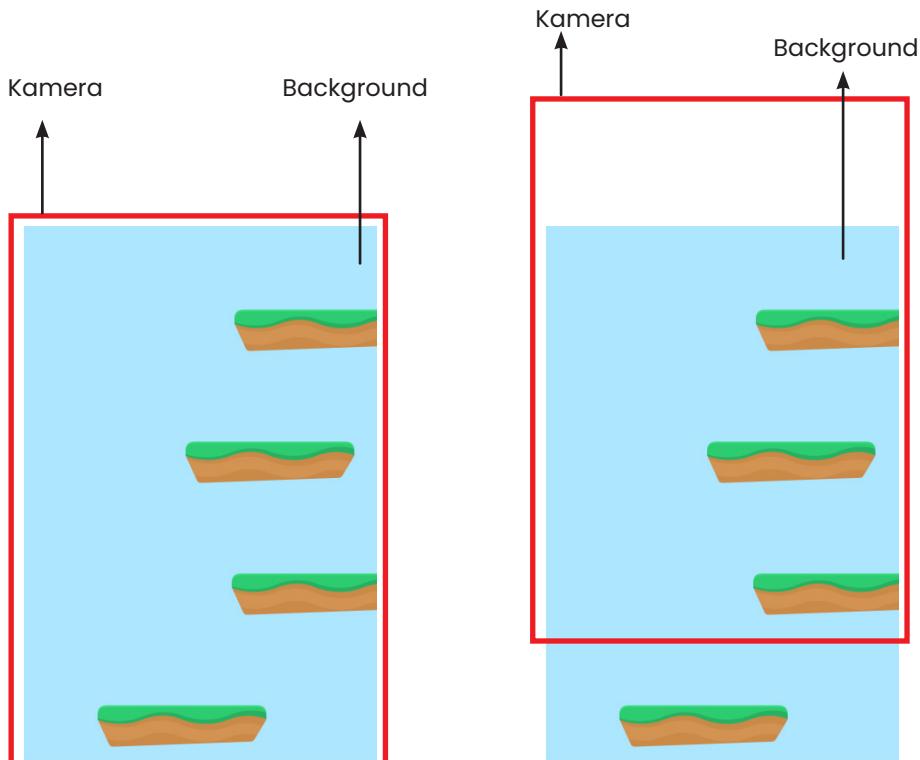


#### SetScrollFactor pada Background



Tahukah kamu jika kamera pada game dapat berpindah-pindah?

Jika objek hanya ditambahkan sekali pada method create() (contohnya background), maka objek itu akan tertinggal diposisi x dan y nya sendiri saat kamera berpindah.



Jika kamera di scroll, maka objek background akan tertinggal diposisi semula (seperti yang terjadi pada game mu sekarang)



Lalu bagaimana caranya agar background tidak tertinggal di bawah saat kamera di scroll ke atas?

Caranya sangat mudah lo! Kita hanya perlu menambahkan kode **setScrollFactor()** !



### Vocabulary

**setScrollFactor()** adalah metode untuk mengatur scroll factor pada suatu objek, apakah objek akan tertinggal saat kamera di scroll atau objek tetap menempel di layout walaupun kamera berpindah.

## Syntax!

**setScrollFactor (x, y)**

Parameter Set Scroll Factor dapat diisi dengan **nilai X saja** atau diisi dengan **nilai X dan Y**. Jika hanya berisi nilai X saja, maka nilai Y sama dengan nilai X.

```
if (y === undefined) { y = x; }
```



Apa fungsi nilai X dan Y?

Nilai **X** berfungsi untuk mengatur scroll factor di posisi **horizontal**

Nilai **Y** berfungsi untuk mengatur scroll factor di posisi **vertical**

**Nilai X dan Y hanya dapat diisi dengan nilai 0 atau 1**

Nilai **0** artinya posisi objek tidak terpengaruh saat kamera di scroll, jadi akan tetap diam pada layout game.

Nilai **1** artinya posisi objek tertinggal di posisi semula saat kamera di scroll, seperti yang terjadi pada game mu sekarang.



## Pertemuan 5-Bunny Jump : Scrolling Platform

### Scrolling Platform

Nah sekarang kita akan membuat **Background** memiliki scroll factor dengan **X = 1** dan **Y = 0**.

Tambahkan kode `setScrollFactor(1, 0)` pada kode **menambahkan background** di method `create()`!



#### Codes

```
this.add.image(240, 320, 'background').setScrollFactor(1, 0)
```

Kita mengatur nilai Y = 0 sehingga background akan tetap menempel pada layout meskipun Player dan Kamera berjalan terus ke atas.

Coba reload dan lihat hasilnya!



#### Challenge!

Coba kamu ganti nilai Y menjadi 1.  
Apa yang terjadi?

Sekarang, coba kamu gerakkan player **ke samping** sejauh-jauhnya!  
Apa yang terjadi pada **layout games**?

Menurutmu, apakah bagus jika sebuah game **tidak memiliki batas** layout? Tentu tidak kan?

Jadi, sekarang ayo kita buat game mu memiliki batas layout.



## 2. Deadzone



Pernahkah kamu main Game Minecraft?

Saat menggerakkan Player, pasti **terdapat area yang tidak bisa dilewati player**.



### Vocabulary

**Deadzone** adalah wilayah yang dapat dijangkau oleh kamera. Deadzone dapat digunakan jika **terdapat target** di dalamnya.

Dalam game ini, kita telah mengatur **target** yang diikuti kamera yaitu objek **Player**.

Tambahkan kode berikut ini pada method **create()**!



### Codes

```
// atur dead zone dengan nilai lebar layout dikalikan 1.5
this.cameras.main.setDeadzone(this.scale.width * 1.5)
```

**this.scale.width** -> kode untuk memanggil ukuran lebar game yang telah diatur pada main.js, yaitu 480



Kenapa dikalikan dengan 1.5?

Tujuannya adalah untuk memberikan ruang tambahan untuk player ketika nanti akan melakukan **wrapping**

Kamu sudah mengatur deadzone agar kamera tetap berada pada area layout. Sekarang kamu juga perlu **mengatur agar player tidak dapat keluar dari area tersebut**.



### 3. Layout Wrap



### Vocabulary

**Layout Wrap** membuat player tidak dapat pergi dari layout, baik dalam arah Horizontal maupun Vertical.



## Pertemuan 5-Bunny Jump : Scrolling Platform Moving Left & Right

Contoh game yang menggunakan Layout Wrap adalah Game **Snake**



Meskipun ular digerakkan keluar layout, tapi ular itu akan muncul lagi di sisi yang berlawanan.

Nah pada game Bunny Jump, kita akan membuat **Horizontal Wrap** yang berfungsi agar **Player kembali ke layout meskipun bergerak ke luar layout melalui sisi kanan maupun kiri**.

Buatlah sebuah method baru bernama **horizontalWrap()** dengan kode berikut ini. Tulislah di bawah method **update()**



### Codes

```
// buat method dengan parameter sprite
horizontalWrap(sprite) {
 const halfWidth = sprite.displayWidth * 0.5
 const gameWidth = this.scale.width
 if (sprite.x < -halfWidth) {
 sprite.x = gameWidth + halfWidth
 } else if (sprite.x > gameWidth + halfWidth){
 sprite.x = -halfWidth
 }
}
```

Kita akan mempelajari apa itu **displayWidth** pada pertemuan selanjutnya ya!



Lalu apa maksud **kondisi if else** pada kode di atas?  
Perhatikan penjelasan berikut ini ya!



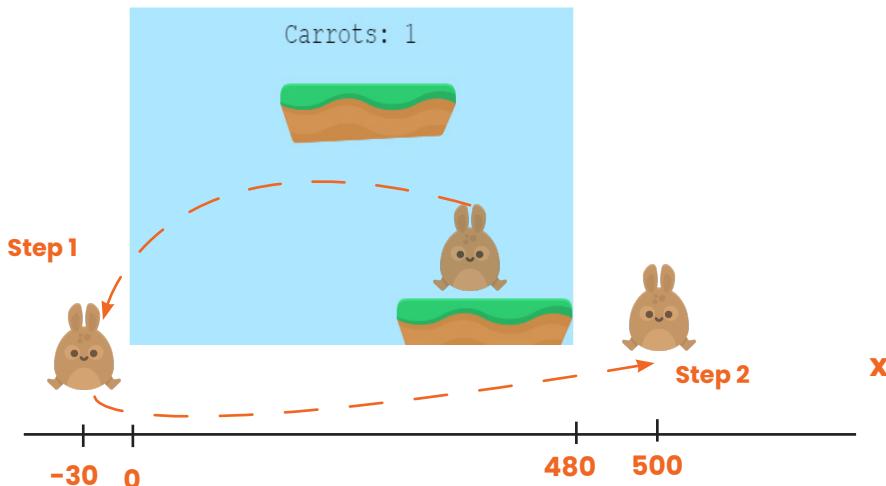
1.

### Untuk kondisi IF

```
if (sprite.x < -halfWidth) {
 sprite.x = gameWidth + halfWidth
}
```

Misalnya :

halfWidth = 20, gameWidth = 480, sprite.x = -30



"Jika Player berada di x = - 30, maka akan dipindah ke x = 500"

2.

### Untuk Kondisi Else If

```
else if(sprite.x > gameWidth + halfWidth){

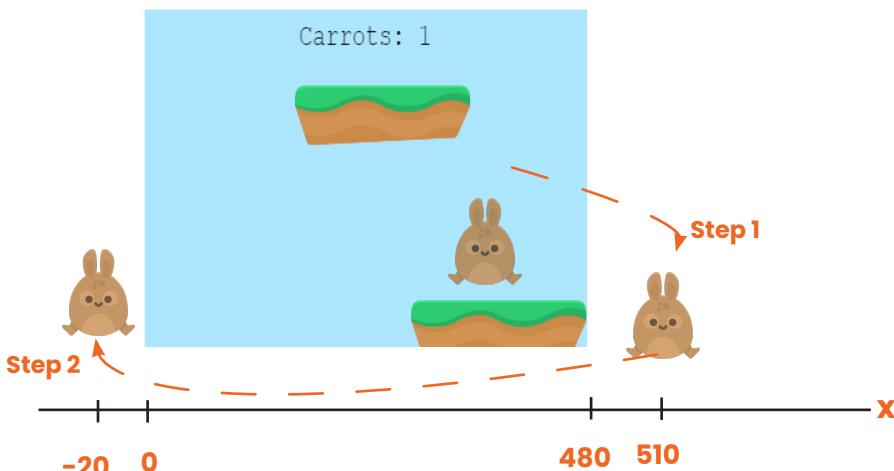
 sprite.x = -halfWidth
}
```

Misalnya :

halfWidth = 20, game width = 480, sprite.x = 510



## Pertemuan 5-Bunny Jump : Scrolling Platform Moving Left & Right



"Jika Player berada di  $x = 510$ , maka akan dipindah ke  $x = -20$ "



Apakah kamu ingat dengan **Function Parameter**?

Pada method **horizontalWrap()** kita menggunakan **parameter sprite**. itu berarti terdapat nilai yang harus dimasukkan saat pemanggilan method horizontalWrap() dan disimpan pada parameter sprite

**horizontalWrap(sprite)**

Sekarang kita perlu memanggil method horizontalWrap() berisi parameter Player, karena Sprite yang kita gunakan adalah Player.



**Do it Your Self !**

Apakah kamu ingat cara memanggil method?

Coba panggil method **horizontalWrap(this.player)** pada method **update()**

Mainkan game mu dan lihat apa yang terjadi ketika player digerakkan ke samping!



Hari/Tanggal :

1. Coba jelaskan dengan kalimatmu, bagaimana konsep iterasi pada platform agar menghasilkan scrolling platform?

.....  
.....  
.....

2. Variabel apa yang harus dideklarasikan dulu agar keyboard dapat digunakan untuk menggerakkan player?

.....  
.....  
.....

3. Apa arti kode dibawah ini?

```
if (cursors.left.isDown && !touchingDown) {
 player.setVelocityX(-200)
```

.....  
.....  
.....

4. Apa itu deadzone? Bagaimana menulis kode untuk mengatur Deadzone pada game Bunny Jump?

.....  
.....  
.....

5. Apa kegunaan method horizontalWrap() ?

.....  
.....  
.....



# 6

# BUNNY JUMP : LET'S COLLECT THE CARROTS



## Apa Yang Akan Kita Pelajari?

1. Membuat Player mengumpulkan Carrots
2. Menampilkan jumlah Carrots yang berhasil dikumpulkan



## BUNNY JUMP



### Warm Up!



Sebelumnya kita telah membuat **Scrolling Platform** dan **membuat player dapat digerakkan dengan keyboard**.



Bisakah kamu menyebutkan **kONSEP** apa saja yang digunakan untuk **membuat Scrolling Platform**?

Apa **kode** yang harus ditambahkan agar **background tidak tertinggal di bawah meskipun player bergerak ke atas**?



Apa yang akan kita tambahkan selanjutnya?

### COLLECTING COINS AND ADD SCORES



Sumber: google

Pernahkah kamu melihat atau bermain game ini?

Game ini bernama **Temple Run** dan **Subway Surfers**

Kedua contoh games itu memiliki **item** yang harus dikumpulkan, yaitu **koin**.

Apabila Player berhasil mengambil koin, maka **nilai koin atau scores akan bertambah**.

**Nah, kita juga akan menambahkan item yang harus dikumpulkan oleh Player pada game Bunny Jump, yaitu Carrot.**



## Collecting Carrots

Ikuti langkah-langkah berikut ini untuk membuat Player dapat mengumpulkan Carrots ya!



### 1. Menambahkan Objek Carrot

Sebelum menambahkan carrots, kita harus membuat **class Carrot terlebih dahulu**.

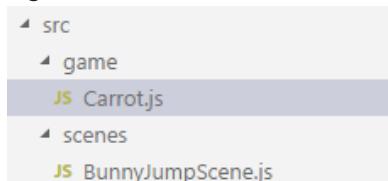


Masih ingat apa itu Class?

Karena kita akan membuat objek baru, jadi kita perlu membuat **template menggunakan Class** agar lebih mudah untuk mengatur propertiesnya nanti!

Cara membuat class Carrot adalah sebagai berikut.

- Klik kanan pada folder src > pilih New Folder > Buat Nama folder “Game”
- Lalu Klik kanan pada folder Game > New File > **Carrot.js**



Kita meletakkannya pada folder baru agar file menjadi lebih rapi dan dikelompokkan jelas mana folder untuk Scene dan mana folder untuk Class

Sekarang tulis kode ini pada kelas **Carrot.js**



#### Codes

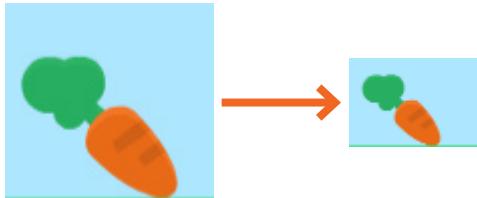
```
import Phaser from 'phaser'

export default class Carrot extends Phaser.Physics.Arcade.Sprite {
 constructor(scene, x, y, texture) {
 super(scene, x, y, texture)
 this.setScale(0.5)
 }
}
```



## Pertemuan 6-Bunny Jump : Let's Collect the Carrots Collecting Carrots

Kita mengatur ukuran Carrot menjadi **0.5** karena ukuran asli yang terdapat pada asset terlalu besar.



**Setelah membuat kelas Carrot sekarang kita akan kembali ke kelas BunnyJumpScene**

Kita akan menambahkan objek carrot sebagai objek group. Kerjakan challenge ini untuk menambahkan carrot ya!



### Do it Your Self !

Penulisan kode seperti ini pernah kita lakukan sebelumnya, jadi pastikan kamu ingat dimana kode berikut ini harus ditulis ya!

1. Import kelas Carrot dari folder game

```
import Carrot from '../game/Carrot.'
```

2. Deklarasikan variabel carrots

```
var carrots
```

3. Tambahkan objek carrot yang di import dari classType Carrot

```
this. carrots = this.physics.add.group({
 classType: Carrot
})
```



Coba **reload** game mu, apakah carrot sudah muncul?  
Mengapa demikian?  
Itu karena kita belum menambahkan posisi **x** dan **y** untuk carrot

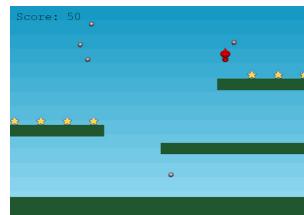


## 2. Membuat Collider Carrot dan Platform

Nah sekarang kita akan membuat carrot berada di atas platform. Caranya adalah dengan menambahkan **collider**.

Ingin dengan game Collecting Star?

Kita sudah pernah membuat **Star berada diatas Platform** menggunakan konsep **collider**



Sekarang kita juga akan membuat **collider untuk Carrot dan Platform**.



### Codes

```
// platforms berbenturan dengan carrots
this.physics.add.collider(this.platforms, this.carrots)
```

Selanjutnya, kita perlu membuat sebuah method baru yang berisi kode untuk mengatur posisi carrot.



## 2. Membuat Function untuk Carrot

Saat membuat method nanti kita akan melihat kode-kode baru yaitu:

- Properti **displayHeight**
- Method **existing()**

### 1. Properti displayHeight

Sebelumnya, kita sudah mengenal properti **width** dan **height**, yaitu ukuran width (lebar) dan height (tinggi) **dari gambar yang di-load ke dalam BunnyJumpScene atau ukuran aslinya**.

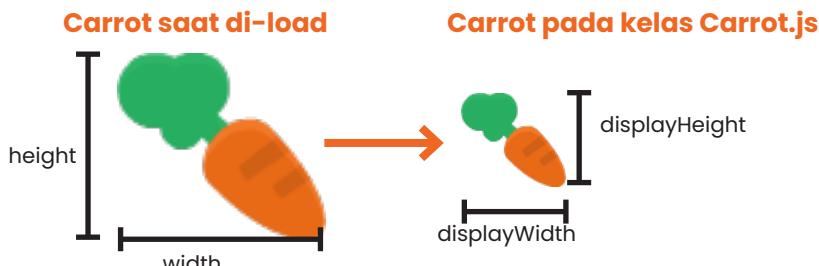


Lalu, apa bedanya dengan **displayHeight** dan **displayWidth**? Yuk kita lihat penjelasan berikut ini!



## Pertemuan 6-Bunny Jump : Let's Collect the Carrots

### Collecting Carrots



Sebelumnya kita telah mengatur ukuran Carrot menjadi lebih kecil dari gambar aslinya menggunakan **setScale(0.5)** pada file **Carrot.js**. Nah, untuk menggunakan properti **width** dari gambar yang telah dikecilkan, kita menggunakan kode **displayWidth**. Begitu juga dengan **displayHeight**.



Bagaimana jika kita menggunakan **width** dan **height**? Maka nanti ukuran width atau height yang ditampilkan adalah **ukuran dari gambar asli**

## 2. Method existing()

Method **existing()** digunakan untuk menambahkan fisik dari game object. Biasanya method ini digunakan setelah menambahkan game object.

### Syntax!

```
this.add.existing(gameObject)
```

Sekarang, Tulis kode ini paling bawah (setelah method horizontalWrap)

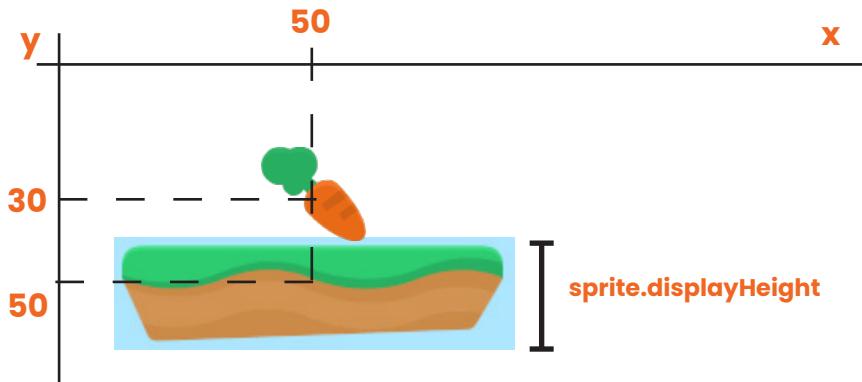


### Codes

```
addCarrotAbove(sprite) {
 const y = sprite.y - sprite.displayHeight
 const carrot = this.carrots.get(sprite.x, y, 'carrot')
 // menambahkan fisik dari carrot.
 this.add.existing(carrot)
 carrot.body.setSize(carrot.width, carrot.height)
 return carrot
}
```



Kenapa `const y = sprite.y - sprite.displayHeight` ?  
Perhatikan penjelasan di bawah ini ya!



Misalnya sprite platform itu berada pada **x = 50, y = 50**  
Dan **displayHeight = 20**

Maka :  $\text{const } y = \text{sprite.y} - \text{sprite.displayHeight}$   
= 50 - 20  
= 30

Jadi, carrot akan diletakkan pada posisi **y = 30**



### Challenge!

Agar lebih paham, coba kamu simulasikan pada link berikut ini.

[tiny.cc/XYSimulation](http://tiny.cc/XYSimulation)

1. Comment line 27 - line 43
2. Edit Line 25 dan line 26 menjadi :
  - line 25 : `this.add.image(50, 50, 'ball').setOrigin(0, 0);`
  - line 26 : `this.add.image(50, 30, 'ball').setOrigin(0, 0.5);`
3. Lalu klik Run Code

Launch

Run Code

Save



## Pertemuan 6-Bunny Jump : Let's Collect the Carrots Collecting Carrots

Selanjutnya panggil method addCarrotAbove() di dalam iterasi platform tepatnya pada tanda komentar!



### Codes

```
this.platforms.children.iterate(child => {
 const platformChild = child
 const scrollY = this.cameras.main.scrollY
 if (platformChild.y >= scrollY + 700) {
 platformChild.y = scrollY - Phaser.Math.Between(50, 100)
 platformChild.body.updateFromGameObject()

 //panggil disini
 this.addCarrotAbove(platformChild)

 }
})
```



Kenapa method addCarrotAbove() dipanggil disana?

Itu karena kita ingin **menambahkan carrot setiap platform baru dibuat.**

Lalu, kode itu berada di method update() bukan?

Artinya **kode itu akan dijalankan terus menerus selama game masih berjalan.**

Sekarang kita akan membuat carrot menghilang dengan kode overlap!



### 3. Membuat Player Overlap Carrot

Kita sudah berhasil membuat carrot muncul di atas platform, sekarang kita akan **membuat Player dapat mengumpulkan Carrot menggunakan overlap.**

Tambahkan kode berikut ini pada method **create()**



### Codes

```
this.physics.add.overlap(
 this.player,
 this.carrots,
 this.handleCollectCarrot,
 undefined,
 this
)
```

Pada overlap, kita menggunakan kode **this.handleCollectCarrot** yang merupakan method yang dipanggil ketika player overlap dengan carrot.

Sebelum membuat method handleCollectCarrot, kita harus mengetahui method baru dari Phaser yaitu **killAndHide()**.



### Vocabulary

**killAndHide()** adalah method yang digunakan untuk menonaktifkan (kill) dan menyembunyikan (hide) game object.

### Syntax!

```
killAndHide(gameObject)
```

Nah sekarang kita akan membuat method handleCollectCarrot(). Tulis kode berikut ini **di bawah method addCarrotAbove()**.



### Codes

```
handleCollectCarrot(player, carrot) {
 this.carrots.killAndHide(carrot)
 this.physics.world.disableBody(carrot.body)
}
```



## Pertemuan 6-Bunny Jump : Let's Collect the Carrots

### Collecting Carrots

Kode diatas artinya **ketika method handleCollectCarrot() dipanggil, maka fisik carrot akan menghilang.**

Coba **reload** game kamu dan kumpulkan Carrots nya!



Apakah carrot hanya muncul sedikit?

Itu terjadi karena **method killAndHide()** tidak secara otomatis memunculkan lagi objek yang sudah dihilangkan

Kita perlu menambahkan kode berikut ini untuk memunculkannya kembali.

Tambahkan kode **dengan tanda komentar** pada method addCarrotAbove()



#### Codes

```
addCarrotAbove(sprite)
{
 const y = sprite.y - sprite.displayHeight
 const carrot = this.carrots.get(sprite.x, y, 'carrot')
 //tambahkan 2 baris kode ini
 carrot.setActive(true) //mengaktifkan carrot
 carrot.setVisible(true) //menampilkan carrot

 this.add.existing(carrot)
 carrot.body.setSize(carrot.width, carrot.height)
 //tambahkan kode di bawah ini
 this.physics.world.enable(carrot)

 return carrot
}
```

Coba **reload** lagi game mu.

Nah sekarang sudah ada banyak carrot yang muncul kan!



## Keeping Track of the Score

Selanjutnya kita akan membuat teks untuk menampilkan jumlah carrot yang sudah terkumpul.

Kita akan membutuhkan variabel untuk menampung nilai carrots.

1. Buat variabel **carrotsCollected** dengan nilai awal 0

```
var carrotsCollected
```

2. Inisialisasi variabel

Pada method `create()` buat kode di bawah ini

```
this.carrotsCollected = 0
```

3. Increment variabel **carrotsCollected**.

Pada method `handleCollectCarrot()` tambahkan kode di ini

```
this.carrotsCollected++
```

Nah sekarang kita harus menampilkan nilai pada variabel `carrotsCollected` pada sebuah teks.

### 1. Menambahkan Teks Score

Buat kode ini pada method `create()`



#### Codes

```
const style = {color: '#000', fontSize:24}
this.add.text(240, 10, 'Carrots: 0', style)
 .setScrollFactor(0)
 .setOrigin(0.5, 0)
```

Sekarang **reload** game kamu dan pastikan teks **Carrot: 0** sudah muncul di layout ya!



## Pertemuan 6-Bunny Jump : Let's Collect the Carrots Collecting Carrots

Perhatikan pada kode menambahkan teks tadi !

 Kenapa teks juga ditambahkan kode **setScrollFactor()** seperti background?  
Masih ingatkah kamu dengan fungsi kode setScrollFactor?  
setScrollFactor() ditambahkan agar **teks Carrots tidak tertinggal** di bawah saat di scroll ke atas



### Challenge!

Coba kamu ubah properti color, fontsize, dan posisi x dan y dari teks sesuai keinginanmu ya!

**Hint :** cari kode warna di <https://htmcolorcodes.com/>



## 2. Mengubah Teks Sesuai Nilai Carrots

Kita perlu mengubah nilai carrots pada teks sesuai dengan Carrots yang berhasil dikumpulkan oleh Player.

Kita harusinisialisasi variabel baru terlebih dahulu untuk menyimpan nilai teks yang berubah-ubah.

Tulis kode **this.carrotsCollectedText()** = di depan kode **this.add.text** pada method **create()**



### Codes

```
const style = {color: '#000', fontSize:24 }
//tambahkan seperti di bawah ini
this.carrotsCollectedText = this.add.text(240, 10, 'Carrots:
0', style)
 .setScrollFactor(0)
 .setOrigin(0.5, 0)
```



Selanjutnya tambahkan kode berikut ini pada method handleCollectCarrot() di bawah increment variabel carrotCollected



## Codes

```
const value = `Carrots: ${this.carrotsCollected}`
this.carrotsCollectedText.text = value
```

Kode baris pertama disebut dengan **String Interpolation**

String                          Value



## Vocabulary

**String Interpolation** pada JavaScript artinya menambahkan nilai variabel pada suatu string.

Contoh :

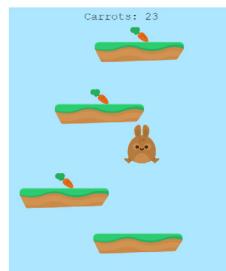
```
const umur = 20
const message = `Umur saya adalah $umur`
message; // => Umur saya adalah 20
```

Jadi, pada kode tadi, nilai variabel **value** adalah **String Interpolation antara String "Carrots:" dan Nilai pada variabel carrotsCollected**

Dan akan ditampilkan pada teks.

**Selesai!**

**Reload game mu dan kumpulkan Carrots sebanyak-banyaknya!**





## Pertemuan 6-Bunny Jump : Let's Collect the Carrots

### Collecting Carrots

Hari/Tanggal :

1. Dimana kita membuat file baru untuk kelas Carrot? Untuk apa kelas Carrot dibuat?

.....  
.....  
.....

2. Bagaimana kode untuk membuat carrot collide dengan platform?

.....  
.....  
.....

3. Apa perbedaan height dan displayHeight pada objek Carrot?

.....  
.....  
.....

4. Method apa yang dibuat untuk menambahkan carrot diatas platform? Bagaimana cara memanggilnya?

.....  
.....  
.....

5. Variabel apa yang digunakan untuk menampung jumlah carrots?

.....  
.....  
.....





# 7

# BUNNY JUMP : HANDLING GAME OVER



Apa Yang Akan Kita Pelajari?

1. Membuat Game Over Scene
2. Menambahkan sound



Pertemuan

7

## BUNNY JUMP

Game Bunny Jump kita hampir selesai!

Sekarang kita perlu menambahkan scene baru yaitu **Game Over Scene**



### Game Over Scene



Sebuah game pasti memiliki kondisi untuk kalah. Sekarang kita juga akan menambahkan kondisi kalah pada game Bunny Jump.

**Kita akan membuat player kalah jika player terjatuh dari platform dan keluar dari layout**

**Untuk membuat Game Over, kita perlu melakukan langkah-langkah ini.**

1. Membuat **Game Over Scene**
2. **Membuat method** untuk menemukan posisi platform paling
3. **Memanggil method** itu ditambah kondisi jika player melewati platform paling bawah
4. Menambahkan Game Over Scene pada **main.js**
5. Menambahkan **Tombol Replay**



## 1. Membuat Game Over Scene

Buatlah sebuah kelas baru untuk mengatur scene Game Over.



Masih ingat cara membuat kelas baru?

Klik kanan pada folder **scenes** > **New File** > ketik **GameOverScene.js**

Selanjutnya tulis kode berikut ini pada kelas GameOverScene



### Codes

```
import Phaser from 'phaser'

export default class GameOverScene extends Phaser.Scene {

 constructor() {
 super('game-over-scene')
 }

 preload() {
 this.load.image('background', 'images/bg_layer1.png')
 //load image teks game over disini
 //load image tombol replay disini
 }

 create() {
 this.add.image(240, 320, 'background')
 this.add.image(240, 280, 'game-over-text')
 }
}
```



## Pertemuan 7-Bunny Jump : Handling Game Over Game Over Scene



### Do it Your Self !

Load gambar teks Game Over dan Tombol Replay pada method preload() dengan nama object di bawah ini !

'game-over-text'

'replay-button'



Replay



## 2. Membuat Method

Kita akan membuat Player kalah saat terjatuh dan tidak mendapat pijakan Platform lagi. Untuk itu, kita harus menghitung posisi platform yang paling bawah terlebih dahulu. **Kembali ke BunnyJumpScene dan Buat method baru setelah method handleCollectCarrot()**



### Codes

```
findBottomMostPlatform() {
 // panggil children dari platforms global
 const platforms = this.platforms.getChildren()
 // ambil item pertama dari array platforms lokal
 let bottomPlatform = platforms[0]
}
```

Pada method findBottomMostPlatform() kita **mendeklarasikan variabel lokal bernama platforms** dan **bottomPlatform**. Kode diatas bisa digambarkan seperti ini.

→ variabel lokal

platforms = [platform1, platform2, platform3, platform4, platform5]

Indeks ke:

0

1

2

3

4

→ bottomPlatform



Platform terus bertambah, lalu bagaimana cara **mencari bottomPlatform selanjutnya**?

Kita bisa menggunakan **for loop untuk membandingkan setiap platform pada array**.

**Tambahkan kode for loop ini pada method findBottomMostPlatform()**



### Codes

```
findBottomMostPlatform() {
 //kode sebelumnya

 for (let i=1; i<platforms.length; i++) {
 const platform = platforms[i]

 }
 return this.bottomPlatform
}
```

Selanjutnya tambahkan **kondisi** di dalam for loop untuk mengecek **mana platform yang posisi Y nya paling besar (posisinya paling bawah)**, maka itu yang menjadi bottomPlatform

**Tambahkan kondisi pada for loop**



### Codes

```
for (let i=1; i<platforms.length; i++) {
 const platform = platforms[i]

 // tambahkan kondisi ini
 if (platform.y < bottomPlatform.y)
 {
 continue
 }
 bottomPlatform = platform
}
```



## Pertemuan 7-Bunny Jump : Handling Game Over Game Over Scene



### 3. Memanggil Method

Untuk menggunakan bottomPlatform yang dibuat pada method findBottomMostPlatform(), kita bisa memanggilnya pada method update()

**Tambahkan kode ini pada method update()**



#### Codes

```
const bottomPlatform = this.findBottomMostPlatform()
if(this.player.y > bottomPlatform.y + 200){
 this.scene.start('game-over-scene')
}
```

Pada method update(). kita menginisialisasi variabel lokal bottomPlatform dengan method findbottomMostPlatform(). Kita juga menambahkan kondisi:

```
if(this.player.y > bottomPlatform.y + 200){
 this.scene.start('game-over-scene')
}
```

Artinya: jika posisi player melewati bottomPlatform + 200, pindah ke game over scene



### 4. Konfigurasi pada Main.js

Seperti yang telah kita lakukan sebelumnya pada BunnyJumpScene, kita juga perlu mendaftarkan GameOverScene pada main.js

1. Import kelas GameOverScene di bawah BunnyJumpScene

```
import GameOverScene from './scenes/GameOverScene';
```

2. Tambahkan GameOver pada properti scene

```
scene: [BunnyJumpScene, GameOver]
```



Nah sekarang kamu sudah bisa menggunakan Game Over pada game Bunny Jump!

**Coba reload game mu dan pastikan game over sudah berhasil dijalankan!**



### Challenge!

Coba kamu ubah nilai **200** pada kondisi ini dengan nilai yang lain sesukamu dan lihat perubahannya

```
if(this.player.y > bottomPlatform.y + 200)
{
 this.scene.start('game-over-scene')}
```



## 5. Membuat Play Again

Game mu hampir sempurna!

Ayo kita tambahkan button Play Again agar bisa memainkannya kembali setelah game over!

**Deklarasikan variabel replayButton terlebih dahulu ya!**



### Codes

```
var replayButton
```



## Pertemuan 7-Bunny Jump : Handling Game Over Game Over Scene

Selanjutnya kita tambahkan replay button pada method `create()`



### Codes

```
this.replayButton = this.add.image(240, 420,
 'replay-button').setInteractive()
// berpindah ke bunny jump scene ketika button di klik
this.replayButton.once('pointerup', () => {
 this.scene.start('bunny-jump-scene')
}, this)
```



Apa itu `setInteractive()`?



### Vocabulary

`setInteractive()` adalah method yang digunakan untuk mengaktifkan properti input pada suatu objek sehingga objek dapat diklik.

**Sekarang coba mainkan game mu dan pastikan semua sudah berjalan dengan benar ya!**



## Adding Sound Effect

Agar game lebih menarik, kita bisa menambahkan efek suara saat player lompat.

Sebelumnya pastikan kamu telah menambahkan sound di folder **public > sfx** ya!

Sama seperti image, sound juga perlu di load pada **preload()**



### Codes

```
this.load.audio('jumpSound', 'sfx/phaseJump1.ogg')
```

Selanjutnya kita panggil sound itu dengan method **this.sound.play()**. pada method **update()**

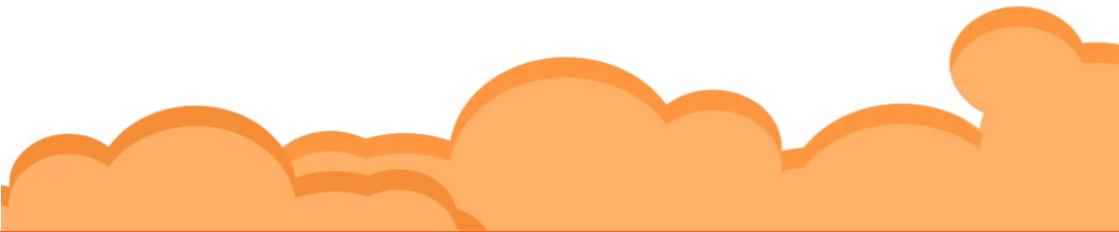


### Codes

```
//kondisi jika player menyentuh bawah
if (touchingDown) {
 this.player.setVelocityY(-300)
 this.player.setTexture('bunny_jump')
 this.sound.play('jumpSound')
```

**SELESAI!**

**Ayo mainkan game mu dan berlomba untuk mengumpulkan Carrots sebanyak-banyaknya dengan temanmu!**





Hari/Tanggal : \_\_\_\_\_

1. Gambar apa saja yg perlu di load untuk membuat Game Over Scene?

.....  
.....  
.....

2. Apa fungsi dari method findBottomMostPlatform() ?

.....  
.....  
.....

3. Apa kegunaan kode ini pada method findBottomMostPlatform() ?

```
for (let i=1; i<platforms.length; i++) {
 const platform = platforms[i] }
```

.....  
.....  
.....

4. Bagaimana cara menambahkan Game Over Scene pada kelas main.js ?

.....  
.....  
.....

5. Apa fungsi dari method setInteractive() pada button replay?

.....  
.....  
.....





8

# CORONA BUSTER : ADD BUTTON AND PLAYER ANIMATION



Apa Yang Akan Kita Pelajari?

1. Membuat Moving Clouds
2. Menambahkan Button Moving Player



## Corona Buster

 Warm Up!

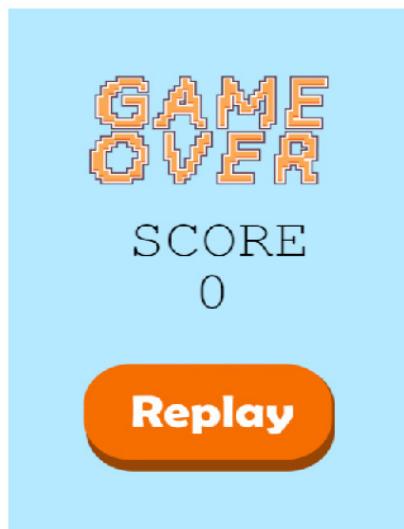
**Ingat dengan game Clone Wars yang kamu buat saat level Beginner?**

Kali ini kita juga akan membuat shooter game seperti itu, tapi tentunya game kali ini akan lebih menantang!

CLONE WARS



## CORONA BUSTER



**Ayo Selamatkan Dunia dari Virus Corona!**

Corona Buster adalah shooter game yang memiliki goal untuk membunuh Virus Corona. Player Ship harus digerakkan untuk menuju ke arah Virus Corona yang datang dan menembak Virus itu dengan Laser.

Player Ship juga bisa mengambil Handsanitizer yang bisa menambahkan nyawanya. Jika Player Ship terkena Virus Corona maka nyawanya akan berkurang dan bisa kalah saat nyawanya telah habis. Yuk kita bantu Player Ship agar membasmi Virus Corona!



**Yuk kita mulai persiapan untuk membuat Game Corona Buster!**



## Preparation

Nah seperti biasa kita harus melakukan beberapa persiapan dulu sebelum mulai coding!



### 1. Persiapan Template

Ingin copy-paste terlebih dahulu file phaser3-parcel-template-master dan rename dengan nama **Corona Buster**. Buat file baru pada folder Scene dengan nama **CoronaBusterScene.js** dan tuliskan kode berikut ini.



## Codes

```
import Phaser from 'phaser'
export default class CoronaBusterScene extends Phaser.Scene
{
 constructor()
 {
 super('corona-buster-scene')
 }
 preload()
 {
 }
 create()
 {
 }
 update()
 {
 }
}
```



## Pertemuan 8-Corona Buster: Moving Clouds and Add Button Preparation

Pada file **index.html**, ubah judul *Phaser3 + Parceljs Template* menjadi **Corona Buster**. Nantinya itu akan mengubah judul tab pada browser

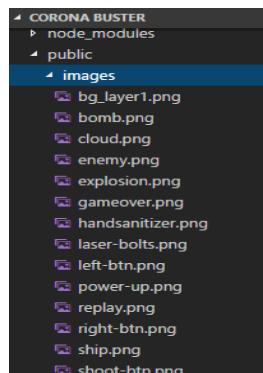


### 2. Download Assets

Kita akan memerlukan assets untuk membuat game Corona Buster. Ingat buat folder public lalu buat juga folder images di dalam folder public dulu ya!

- Download assets di link ini:  
[http://tiny.cc/meeting8\\_assets](http://tiny.cc/meeting8_assets)

- **Copy paste** semua assets gambar pada folder images



### 3. Configuration

Selanjutnya coba lakukan konfigurasi ini sendiri ya! Caranya sama dengan meeting yang sebelumnya. Pada kelas main.js :

- Atur ukuran layout game  
**width** =400 dan **height** = 620

- Hapus properti **gravity** ini karena kita tidak memerlukannya  
arcade: {  
    gravity: { y: 200 },  
}

- Import CoronaBusterScene dan tambahkan pada properti scene



Coba jalankan game kamu dan pastikan layout game berbentuk Portrait!

Buka terminal atau cmd > ketik npm run start



## Menambahkan Background dan Clouds



### 1. Background

Background yang akan digunakan harus di-load lalu juga ditambahkan.



Masih ingat cara load dan add image background?  
Lihat kode di bawah ini! **Dimana kita harus menulisnya?**



### Codes

```
this.load.image('background', 'images/bg_layer1.png')
```

Lalu kita akan menambahkan background pada posisi tengah-tengah layout game. Buat kode ini pada method `create()`



### Codes

```
const gameWidth = this.scale.width * 0.5
const gameHeight = this.scale.height * 0.5
this.add.image(gameWidth, gameHeight, 'background')
```

**Nah sekarang coba reload dan pastikan background sudah berhasil di munculkan dengan benar!**



Tapi, layout game berada di kiri kan?

Sekarang ayo kita atur agar layout game berada ditengah-tengah layar sehingga game terlihat lebih bagus!



## Pertemuan 8-Corona Buster: Moving Clouds and Add Button

### Add Clouds

Buka kelas main.js dan buat kode berikut ini setelah properti physics!

#### Codes

```
// kode physics disini,
scale: {
 mode: Phaser.Scale.FIT,
 autoCenter: Phaser.Scale.CENTER_BOTH
},
// kode scene disini
```

Kode di atas akan mengatur layout game agar berada di tengah-tengah layar (center).

Coba lihat perubahannya!



## 2. Clouds

Clouds adalah awan yang juga menjadi background game dan akan bergerak ke arah vertikal.

Untuk membuat clouds yang bergerak, ayo kita lakukan langkah-langkah ini!

1. Inisialisasi variabel clouds
2. Load image clouds dan Add physics group
3. Iterasi Clouds agar bisa bergerak



## 1. Inisialisasi Variabel Cloud

Kali ini kita akan menginisialisasi semua variabel dengan cara yang berbeda dari sebelumnya. Variabel akan di deklarasikan sekaligus diberi nilai pada method `init()`



Apa itu method `init()` ?



### Vocabulary

**Init()** berasal dari kata initialize yang berarti inisialisasi. Sesuai namanya, method `init()` adalah tempat untuk menginisialisasi property dengan sebuah nilai awal.

Jadi, sejak scene dieksekusi, property langsung diberi nilai awal.



Lalu, apa bedanya dengan deklarasi variabel yang kita lakukan pada game Bunny Jump?

```
var player
```

Itu hanya **mendeklarasikan variabel dengan nilai default undefined**. Jika ingin mengisi variabel yang di awalnya memiliki nilai tertentu, maka dapat di inisialisasi pada method `init()`.



## Pertemuan 8-Corona Buster: Moving Clouds and Add Button

### Moving Clouds

Buat **method init()** di bawah method constructor() dan inisialisasi properti clouds undefined agar setiap game dimulai properti clouds nilai awalnya tidak didefinisikan dulu.



#### Codes

```
// method constructor disini

init() {
 this.clouds = undefined
}
```

Nah kita sudah menginisialisasi properti **clouds**, sekarang kita perlu **load** dan **add image clouds** agar bisa ditampilkan ke layout game.

## 2. Load dan Add Image Cloud

Sebelumnya kamu sudah pernah melakukan load image pada game Bunny Jump, Coba sekarang load image cloud mu sendiri ya!



#### Do it Your Self !

Load image cloud !  
(**'cloud'**, '**images/cloud.png**' )

Selanjutnya kita akan menambahkan image cloud pada method **create()**.

Cloud bersifat dinamis (dapat bergerak) sehingga kita harus menambahkannya ke dalam group



Tulis kode berikut ini di dalam method **create()** !



## Codes

```
this.clouds = this.physics.add.group({
 key: 'cloud',
 //ulangi tampilkan awan
 repeat: 20
})
Phaser.Actions.RandomRectangle(this.clouds.getChildren(),
this.physics.world.bounds);
```



Apa itu Random Rectangle?

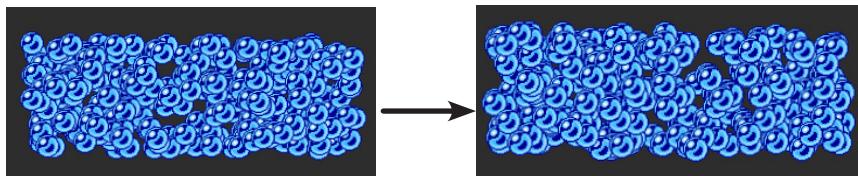


## Vocabulary

Method **RandomRectangle** digunakan untuk mengatur posisi acak dalam bentuk rectangle (persegi panjang)

**Kamu bisa cek simulasinya pada link ini agar lebih paham ya!**  
[tiny.cc/RandomRectangle](http://tiny.cc/RandomRectangle)

Coba kamu klik **Run Code** dan perhatikan posisi objek itu. Lakukan beberapa kali dan kamu akan melihat **posisi objek-objek itu akan berubah setiap kali di Run**



Itu karena kita menggunakan **RandomRectangle** sehingga posisi objek akan berubah-ubah



Sama halnya dengan Cloud yang kita buat, posisi nya akan ditampilkan secara acak dan berubah-ubah.



### Challenge!

Coba kamu ubah **repeat: 20** menjadi angka yang lainnya dan lihat apa yang terjadi!



Tapi, apa awan mu sudah bergerak ke bawah?

Untuk membuat awan bisa bergerak, kita akan memerlukan konsep iterasi

### 3. Iterasi pada Clouds

Selanjutnya buat iterasi child pada method update seperti kode di bawah ini.



### Codes

```
update(time)
{
 this.clouds.children.iterate((child) => {
 //arah gerak awan ke bawah
 child.setVelocityY(20)

 if (child.y > this.scale.height){
 child.x = Phaser.Math.Between(10, 400)
 child.y = child.displayHeight * -1
 }
 })
}
```

Kondisi pada kode diatas akan menyebabkan awan berpindah ke atas lagi saat sudah melebihi batas layout.  
Sekarang coba lihat hasilnya pada layout mu!



## Menambahkan Button



Button yang kita tambahkan akan digunakan untuk menggerakkan player ke kanan dan ke kiri dan juga untuk menembak.

Pertama kita juga harus menginisialisasikan properti untuk button left, right, dan shoot.

Tulis kode ini pada method **init()**



### Codes

```
this.nav_left = false
this.nav_right = false
this.shoot = false
```



Kenapa kita mengisinya dengan nilai False ?

Karena kita ingin membuat button tidak aktif dulu saat permainan dimulai, tapi kita akan mengaktifkannya nanti saat button di klik.

Ayo sekarang latihan lagi untuk load image button mu sendiri ya!



### Do it Your Self !

Load image button left-btn, right-btn, dan shoot-btn !

Contoh:

```
('left-btn' 'images/left-btn.png')
```



### Add Button

Nah selanjutnya kita akan membuat method baru untuk memunculkan button pada layout



#### 1. Membuat Method createButton()

Sebelum membuat kode untuk method createButton(), kita akan berkenalan dulu dengan beberapa method dari phaser ini.

1. **addPointer**, yaitu method bawaan phaser yang berasal dari kelas InputPlugin untuk menambahkan objek Input seperti Mouse.

**Syntax** : `addPointer(quantity)`

**Fungsi** : menambahkan pointer baru dengan parameter quantity atau jumlah pointer yang ingin ditambahkan

**Contoh** : `addPointer(3)` -> artinya menambahkan 3 pointer

2. **setDepth**, yaitu method bawaan phaser yang berasal dari namespaces GameObject.Components.Depth

**Syntax** : `setDepth(value)`

**Fungsi** : mengatur layer atau z order objek. Semakin besar nilainya, semakin di depan posisi objeknya

**Contoh** : cloud tidak memanggil setDepth()  
enemy memanggil method setDepth(1)



Maka enemy akan berada lebih  
di depan dibandingkan dengan  
cloud

3. **setAlpha**, yaitu method bawaan phaser yang berasal dari namespaces GameObject.Components.Alpha

**Syntax** : `setAlpha(topLeft, topRight, bottomLeft, bottomRight)`

**Fungsi** : mengatur transparansi (opacity) object

**Contoh** : `setAlpha(0.5)` -> artinya opacity dari semua sisi





DID YOU  
KNOW?

**Namespace** adalah istilah dalam pemrograman untuk mengelompokkan beberapa identifier ke dalam kelompok yang sama.

Nah sekarang ayo kita gunakan method tadi pada method `createButton()`.

### Codes

```
createButton()
{
 this.input.addPointer(3)

 let shoot = this.add.image(320, 550, 'shoot-btn')
 .setInteractive().setDepth(0.5).setAlpha(0.8)
 let nav_left = this.add.image(50, 550, 'left-btn')
 .setInteractive().setDepth(0.5).setAlpha(0.8)
 let nav_right = this.add.image(nav_left.x + nav_left
 .displayWidth + 20, 550, 'right-btn')
 .setInteractive().setDepth(0.5).setAlpha(0.8)

}
```

Pada kode diatas kita menambahkan 3 input pointer. Lalu kita menambahkan gambar yang disimpan pada variabel lokal yaitu **shoot, nav\_left, nav\_right**

Pada inisialisasi properti shoot, nav\_left, dan nav\_right kita memanggil 3 method yaitu :

- **setInteractive()** -> agar gambar dapat diklik
- **setDepth(0.5)** -> mengatur posisi button di depan awan
- **setAlpha(0.8)** -> mengatur opacity menjadi 0.8



## Pertemuan 8-Corona Buster: Add Button and Player

### Add Button

Coba kamu perhatikan pada kode pada nav\_right !



Mengapa nilai x dari nav\_right ditulis dengan cara yang rumit?  $\rightarrow \text{nav_left.x} + \text{nav_left.displayWidth} + 20$

**Sebenarnya, ini justru cara yang mudah untuk menampilkan button right simetris dengan button left daripada mengira-ngira dengan sebuah angka**



#### Challenge!

Coba kamu **hapus kode : + 20** pada kode tadi sehingga menjadi : `nav_left.x + nav_left.displayWidth`

Hasilnya seperti gambar di samping bukan?



Lalu kamu **tambahkan lagi kode : + 20** atau angka berapapun sesukamu.  
Maka hasilnya adalah memberikan jarak antara button left dan right





Nah kita baru menginisialisai properti button saja, sekarang kita perlu membuat kode agar bisa diaktifkan.

Lanjutkan kode pada `createButton()` tadi dengan kode ini!



### Codes

```
nav_left.on('pointerdown', () => { this.nav_left = true }, this)
nav_left.on('pointerout', () => { this.nav_left = false}, this)
nav_right.on('pointerdown', () => { this.nav_right = true }, this)
nav_right.on('pointerout', () => { this.nav_right = false}, this)
shoot.on('pointerdown', () => { this.shoot = true }, this)
shoot.on('pointerout', () => { this.shoot = false}, this)
```

Bisa dilihat pada setiap button, kita menggunakan 2 jenis event yaitu `pointerdown` dan `pointerup`

Ketika `pointerup` (diklik) maka jalankan pemanggilan kode pada button itu (true)

Ketika `pointerdown` (tidak di klik) maka kode pada button itu tidak dijalankan (false)

Selanjutnya, panggil method `createButton()` pada method `create()`



### Codes

```
this.createButton()
```



### Add Button

Hari/Tanggal :

1. Apa saja image yang ditambahkan kali ini?

.....  
.....  
.....

2. Apa fungsi kode ini yang ditulis di main.js?

```
scale: {
 mode: Phaser.Scale.FIT,
 autoCenter: Phaser.Scale.CENTER_BOTH }
```

.....  
.....

3. Apa fungsi method init() ?

.....  
.....  
.....

4. Clouds dimunculkan dengan kode Random Rectangle. Apa itu Random Rectangle?

.....  
.....  
.....

5. Variabel apa saja yang di deklarasikan pada method createButton() ?

.....  
.....  
.....





# References



- Tynker Coding and STEAM Curriculum
- Google's Teachable Machine
- Common Sense Education
- W3School
- Phaser.io