

JavaScript Fundamentals

1. History JavaScript

HTML dan CSS adalah bahasa markup. Bahasa markup digunakan untuk mendeskripsikan dan mendefinisikan elemen-elemen di dalam dokumen. JavaScript adalah bahasa pemrograman. Bahasa pemrograman digunakan untuk mengomunikasikan instruksi ke mesin. Bahasa pemrograman dapat digunakan untuk mengontrol perilaku mesin dan mengekspresikan algoritme.

JavaScript pertama kali dikembangkan oleh Brendan Eich pada tahun 1995 ketika ia bekerja di Netscape Communications Corporation. Awalnya dikenal dengan nama Mocha, kemudian LiveScript, dan akhirnya disebut JavaScript. Tujuannya adalah untuk memberikan interaktivitas pada halaman web. Saat ini, JavaScript adalah salah satu bahasa pemrograman paling populer di dunia, digunakan untuk mengembangkan aplikasi web, aplikasi seluler, permainan, dan banyak lagi.

2. The JavaScript Console

Konsol JavaScript adalah alat debugging yang tersedia di browser web modern seperti Google Chrome, Mozilla Firefox, dan lain-lain. Konsol ini memungkinkan Anda untuk mengevaluasi kode JavaScript, memeriksa variabel, memecahkan masalah, dan melakukan debugging pada halaman web. Anda dapat membukanya dengan menekan tombol F12 pada keyboard atau klik kanan pada halaman web dan pilih opsi "Inspect" atau "Inspect Element".

3. Developer Tools on Different Browsers

- Chrome DevTools adalah sekumpulan alat pembuat web dan debug yang terpasang di Google Chrome. Gunakan DevTools untuk mengulangi, men-debug, dan membuat profil situs Anda.
- Alat Pengembang Firefox memungkinkan Anda memeriksa, mengedit, dan men-debug HTML, CSS, dan JavaScript di desktop dan seluler. Selain itu, Anda dapat mengunduh versi Firefox yang disebut Edisi Pengembang Firefox yang disesuaikan untuk pengembang, menampilkan fitur Firefox terbaru dan alat pengembang eksperimental.
- Jika Anda menggunakan Internet Explorer, Anda dapat mengakses alat pengembang F12 hanya dengan menekan F12. Fitur bervariasi antar versi, tetapi mulai dari Internet Explorer 8 tetap cukup konsisten.
- Microsoft Edge memperkenalkan peningkatan baru yang hebat pada alat pengembang F12 yang terlihat di Internet Explorer. Alat baru dibangun di TypeScript, dan selalu

berjalan, jadi tidak perlu memuat ulang. Selain itu, dokumentasi alat pengembang F12 kini tersedia sepenuhnya di GitHub .

- Untuk setiap pengguna Mac, Safari menyertakan Web Inspector, alat canggih yang memudahkan untuk memodifikasi, men-debug, dan mengoptimalkan situs web untuk performa puncak dan kompatibilitas di kedua platform.
- Opera adalah peramban web yang cepat, ramping, dan andal.

4. console.log

console.logdigunakan untuk menampilkan konten ke konsol JavaScript. Contoh Jalankan kode berikut di konsol:

code: console.log("hiya friend!");


Cetakan: "hai teman!"

5. Demo JavaScript

```
document.getElementsByTagName("h1")[0].style.color = "#ff0000"
```

Pertanyaan Kuis

Apa yang terjadi saat Anda menjalankan baris kode tersebut di konsol JavaScript?

- ☐ Judulnya menghilang
- ☒ Heading berubah menjadi merah 
- ☐ Semua teks di halaman berubah menjadi merah
- ☐ Tidak terjadi apa-apa

Kirim

memodifikasi CSS. Apa yang membuat javascript begitu istimewa dalam hal ini? Muat ulang halaman, lalu tempel baris kode ini di konsol JavaScript.

```
document.body.addEventListener('click', function () {  
  var myParent = document.getElementsByTagName("h1")[0];  
  var myImage = document.createElement("img");  
  myImage.src = 'https://thecatapi.com/api/images/get?format=src&type=gif';  
  myParent.appendChild(myImage);  
  myImage.style.marginLeft = "160px";  
});
```

Jika Anda bingung karena tidak ada yang terjadi. Jangan khawatir. Klik di suatu tempat di halaman untuk melihat efeknya. Anda dapat me-refresh halaman untuk mengembalikan halaman ke keadaan aslinya.

Pertanyaan Kuis

Apa yang terjadi ketika Anda menjalankan kode dari atas di konsol JavaScript *dan kemudian mengklik halaman tersebut* ?

- ☐ Heading berubah menjadi merah
- ☐ Sebuah spanduk terbang melintasi layar
- ☒ Gambar ditambahkan ke halaman
- ☐ Tidak terjadi apa-apa



Kirim

6. Summary

Ringkasan dibangun di dalam mesin JavaScript, artinya bahwa konstruksi sintaksis ringkasan tersebut adalah fitur bawaan dari bahasa pemrograman JavaScript yang sudah terintegrasi ke dalam mesin JavaScript itu sendiri. Oleh karena itu, kita dapat menggunakan ringkasan di dalam kode JavaScript tanpa perlu menginstal atau menambahkan pustaka atau library tertentu. Fitur ini memudahkan pengembang dalam membuat kode JavaScript dengan lebih efisien dan mudah dibaca. Selain itu, penggunaan ringkasan juga membantu meminimalkan jumlah kode yang perlu ditulis, sehingga dapat meningkatkan produktivitas pengembang.

7. Intro to Data Types

Tipe data adalah konsep penting dalam pemrograman yang mengacu pada jenis nilai yang dapat disimpan dan diolah oleh program komputer. Dalam bahasa pemrograman, tipe data digunakan untuk memastikan bahwa variabel atau nilai yang digunakan dalam program memiliki format yang benar dan dapat diolah dengan benar oleh program.

Dalam bahasa pemrograman JavaScript, terdapat beberapa tipe data dasar yang meliputi: Number: tipe data number digunakan untuk merepresentasikan angka, baik itu bilangan bulat atau desimal.

- String: tipe data string digunakan untuk merepresentasikan teks atau karakter.
- Boolean: tipe data boolean hanya memiliki dua nilai, yaitu true atau false.
- Null: tipe data null digunakan ketika suatu variabel tidak memiliki nilai.

- Undefined: tipe data undefined digunakan ketika suatu variabel belum diberikan nilai.
- Object: tipe data object digunakan untuk merepresentasikan objek dalam JavaScript.
- Array: tipe data array digunakan untuk merepresentasikan kumpulan nilai dalam satu variabel.

8. Angka

Mendefinisikan angka dalam JavaScript sebenarnya cukup sederhana. Tipe data Angka mencakup bilangan bulat positif atau negatif, serta desimal. Memasukkan nomor ke konsol akan mengembalikannya langsung kepada Anda.

Code 3

Pengembalian: 3

Operasi aritmatika Anda juga dapat melakukan perhitungan dengan angka dengan cukup mudah. Pada dasarnya ketikkan ekspresi seperti yang Anda ketikkan di kalkulator.

Code 3 + 2.1

Pengembalian: 5.1

Pertanyaan Kuis

Masukkan ekspresi (satu per satu) ke dalam konsol dan tentukan apa yang dievaluasi setiap ekspresi.

 Ini adalah pertandingan yang benar.

Ekspresi	Larutan
2 + 10 - 19 + 4 - 90 + 1	-92
-20 + -19 - (-10) - (-1) + 24	-4
(10/5) * 4 - 20	-12
4096% 12	4

Operator	Arti
<	Kurang dari
>	Lebih besar dari
<=	Kurang dari atau Sama dengan
>=	Lebih dari atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

Pertanyaan Kuis

Masukkan ekspresi (satu per satu) ke dalam konsol dan tentukan apa yang dievaluasi setiap ekspresi.

✅ Ini adalah pertandingan yang benar.

Ekspresi	Larutan
43 > 47	PALSU
12 == 17	PALSU
3 <= 3	BENAR
1 != 0	BENAR

TIP: Nilai `true` dan `false` sangat penting dalam JavaScript. Nilai-nilai ini disebut **Boolean** dan merupakan tipe data lain dalam JavaScript. Nanti dalam pelajaran ini, Anda akan belajar lebih banyak tentang mengapa Boolean sangat penting dalam pemrograman.

9. Nomor dan Operator – Latihan

Isi bagian yang kosong (1 dari 4)

Lengkapi ekspresi dengan mengisi angka yang hilang. Ekspresi harus sama `20`.

`1 + 1 + 2 + 3 + 5 + [BLANK]`

Pengembalian: 20

Isi bagian yang kosong (2 dari 4)

Lengkapi ekspresi dengan mengisi angka yang hilang. Ekspresi harus sama .

10 == [BLANK]

10

Isi bagian yang kosong (3 dari 4)

Lengkapi ekspresi dengan mengisi operator yang hilang. Ekspresi harus sama .

12 [BLANK] (-5) * 5

Pengembalian: 37

12 + (-5) * 5 + 50

Isi bagian yang kosong (4 dari 4)

Lengkapi ekspresi dengan mengisi operator yang hilang. Ekspresi harus sama .

3 * 4 [BLANK] 5

Pengembalian: 2

3*4-5+5=2

10. Comments

Anda dapat menggunakan komentar untuk membantu menjelaskan kode Anda dan membuatnya lebih jelas. Dalam JavaScript, komentar ditandai dengan garis miring ganda `//`. Apa pun yang ditulis pada baris yang sama setelahnya `//` tidak akan dieksekusi atau ditampilkan. Agar komentar mencakup beberapa baris, tandai awal komentar Anda dengan garis miring dan bintang, lalu sertakan komentar Anda di dalam bintang dan garis miring `/*...*/`.

11. Quiz: First Expression (2-1)

```
1- /*
2  * Programming Quiz: First Expression (2-1)
3  * Write an expression that uses at least three, distinct, arithmetic operators
4  * to log the number 42 to the console.
5  */
6
7- /*
8  * QUIZ REQUIREMENTS
9  * 1. Your code should print the value `42`
10 * 2. You should use at least 3 distinct operators. (`+`, `-`, `*`, `/`, or `%`)
11 * 3. Your code should not be empty
12 */
13
14
15 // this expression equals 4, change it to equal 42
16 console.log((4 + 17) * 2 - 8 % 5 + 3);
```

42

12. String

String pada JavaScript adalah tipe data yang digunakan untuk merepresentasikan teks atau karakter. String pada JavaScript dapat dibuat dengan menggunakan tanda kutip tunggal (`'`) atau ganda (`"`), atau dengan menggunakan backtick (```) untuk membuat template literals. Contohnya `'h'`, `"h"`, `"123"`.

13. String Concatenation

String adalah kumpulan karakter yang diapit di dalam tanda kutip ganda atau tunggal. Anda dapat menggunakan string untuk merepresentasikan data seperti kalimat, nama, alamat, dan lainnya.

Pertanyaan Kuis

Dengan apa hasilnya `"hello" + "world"`?

☐ "Halo Dunia"

☒ "Halo Dunia"



Kirim

Pertanyaan Kuis

Menurut Anda apa yang akan terjadi saat Anda mengetik `"Hello" + 5*10` di konsol JavaScript?

☐ "Halo5*10"

☐ "Halo50"

☒ "Halo + 5*10"



☐ Sebuah kesalahan!

Kirim

Pertanyaan Kuis

Menurut Anda apa yang akan terjadi saat Anda mengetik `"Hello" + 5*10` di konsol?

☐ "Halo5*10"

☒ "Halo50"



☐ "Halo + 5*10"

☐ Sebuah kesalahan!

Kirim

14. Variabel

Variabel pada JavaScript adalah wadah untuk menyimpan nilai atau referensi ke nilai. Variabel dapat dideklarasikan dengan menggunakan kata kunci `let`, `const`, atau `var`.

Contohnya:

```
let nama = "John";
```

```
let umur = 25;
```

```
const PI = 3.14;
```

```
const hariDalamSeminggu = 7;
```

```
var x = 5;
```

```
var y = "Hello";
```

Pertanyaan Kuis

Manakah dari berikut ini nama variabel yang bagus?

☐ `var benda = 1;`

☒ `jumlah var = 1;`



☒ `var postLiked = false;`



☐ `var nama depan = "Richard";`

Kirim

15. Mengonversi Suhu (2-2)

```
9 - /*
10  * QUIZ REQUIREMENTS
11  * 1. Your code should have a variable `celsius`
12  * 2. You code should have a variable `fahrenheit`
13  * 3. Your variable `celsius` should equal `12`
14  * 4. Your variable `fahrenheit` should produce the output equal `53.6`
15  * 5. Your variable `fahrenheit` declaration should use the `celsius` variable
16  * 6. Your variable `fahrenheit` should have the correct formula
17  * 7. Your code should log the `fahrenheit` variable
18  * 8. Your code should not be empty
19  */
20
21
22 // var celsius = 12;
23 // var fahrenheit = /* convert celsius to fahrenheit here */
24
25 // console.log(/* print out result here */);
26
27 let celsius = 12;
28 let fahrenheit = celsius * (9/5) + 32;
29
30 console.log(`${celsius}°C setara dengan ${fahrenheit}°F`);
31
35
```

12°C setara dengan 53,6°F

16. String Index

Indeks pada string di JavaScript dimulai dari 0 dan diurutkan dari kiri ke kanan. Dalam string, setiap karakter dapat diakses dengan menggunakan indeksinya. Indeks pada string pada JavaScript dapat diakses dengan menggunakan tanda kurung siku [].

Contohnya:

```
var name = "James";  
name[0];
```

Quiz Question

What character will be printed to the JavaScript console after running the following lines of code.

```
var quote = "Stay awhile and listen!";  
console.log(quote[6]);
```

☐ a

☒ w



☐ !

☐ h

Submit

17. Escaping Strings

Kode	Karakter
\\	\ (garis miring terbalik)
"	" (kutipan ganda)
'	' (kutipan tunggal)
\N	garis baru
\T	tab

Pertanyaan Kuis

Pilih string yang mengembalikan output berikut:

"The file located at "C:\\Desktop\\My Documents\\Roster\\names.txt" contains the names on the roster."

☐

"The file located at "C:\\Desktop\\My Documents\\Roster\\names.txt" contains the names on the roster."

☐

"The file located at "C:\\Desktop\\My Documents\\Roster\\names.txt\\" contains the names on the roster."

☒

"The file located at "C:\\\\Desktop\\\\My Documents\\\\Roster\\\\names.txt\\" contains the names on the roster."



Kirim

18. Comparing Strings

Cara lain untuk bekerja dengan string adalah dengan membandingkannya. Anda telah melihat operator perbandingan `==` dan `!=` saat Anda membandingkan angka untuk kesetaraan. Anda juga dapat menggunakannya dengan string! Sebagai contoh, mari bandingkan string "Yes" dengan "yes".

19. Quiz: Favorite Food (2-3)

```
1 /*  
2  * Programming Quiz: Favorite Food (2-3)  
3  */  
4  
5 /*  
6  * QUIZ REQUIREMENTS  
7  * 1. Your code should print a string with your favorite food  
8  * 2. Your code should have the first character capitalized  
9  * 3. Your code must have a `console.log()` function  
10 * 4. Your code should not be empty  
11 */  
12 let makananFavorit = "Nasi Goreng";  
13 makananFavorit = makananFavorit.charAt(0).toUpperCase() + makananFavorit.slice(1);  
14  
15 console.log(makananFavorit);
```

Nasi goreng

20. String Equality for All (2-4)

```
1 - /*
2  * Programming Quiz: String Equality for All (2-4)
3  *
4  * QUIZ REQUIREMENTS
5  * 1. Your code should have a variable `answer`
6  * 2. Your code should have "ALL Strings are CrEaTeD equal" on the left side
7  * 3. Your code should use `==` comparison operator
8  * 4. The comparison should evaluate to true
9  * 5. The right side of your expression should match the left side
10 */
11
12 // fix the right side of the expression
13 var answer = "All STRINGS are CrEaTeD Equal".toLowerCase() == "ALL Strings are CrEaTeD equal".toLowerCase();
14 console.log(answer);
```

BENAR

21. Quiz: All Tied Up (2-5)

Kode Anda:

```
1 - /*
2  * Programming Quiz: All Tied Up (2-5)
3  *
4  * Quiz Requirements
5  * 1. Your code should have a variable joke
6  * 2. Your joke should use only one string
7  * 3. Your joke should match the expected format
8  */
9  var leluconSepatu = "Mengapa sepatu-sepatu itu tidak bisa keluar dan bermain? \nKarena mereka semua \"diikat\"!";
10 console.log(leluconSepatu);
11
12
```

Mengapa sepatu-sepatu itu tidak bisa keluar dan bermain? Karena mereka semua "diikat"!

22. Quiz: Yosa Buson (2-6)

```
1  /*
2   * Programming Quiz: Yosa Buson (2-6)
3   */
4
5  /*
6   * QUIZ REQUIREMENTS
7   * 1. Your code should have a variable `haiku`
8   * 2. Your code should use string concatenation
9   * 3. Your poem should match the famous haiku poem
10  */
11
12  var arahAngin = "barat";
13  var keadaanDaun = "benguguran";
14  var lokasiDaun = "timur";
15  var kalimat = "Hembusan angin dari arah ${arahAngin}\nDaun-daun ${keadaanDaun}berkumpul\nDi ${lokasiDaun}"
16  console.log(kalimat);
17
```

Hembusan angin dari arah \${arahAngin} Daun-daun \${keadaanDaun}berkumpul Di \${lokasiDaun}

[Reset Quiz](#)[Test Run](#)[Submit Answer](#)

23. Booleans

Boolean adalah tipe data dalam pemrograman yang hanya memiliki dua nilai, yaitu true atau false. Fungsi dari boolean adalah untuk menyatakan nilai kebenaran dari suatu pernyataan atau kondisi.

24. Quiz: Facebook Post (2-7)

✅ Ini adalah pertandingan yang benar.

Surat	Jenis
A - Keberanian	Rangkaian
B - Temui Christian Plagemann, pemimpin tim untuk program VR Developer Nanodegree baru di Udacity! Di sini dia memperkenalkan dan menjelaskan penawaran terbaru kami	Rangkaian
C - blog.udacity.com	Rangkaian
D - Apakah sesuatu itu "disukai" atau tidak.	Boolean
E - 80	Nomor

[← Sebelumnya](#)[Berikutnya →](#)[📧 Kirim Umpan Balik Halaman](#)

25. Null, Undefined, and NaN

NaN singkatan dari "Not-A-Number" dan sering dikembalikan menunjukkan kesalahan dengan operasi angka. Misalnya, jika Anda menulis beberapa kode yang melakukan perhitungan matematika, dan perhitungan gagal menghasilkan angka yang valid, NaN mungkin dikembalikan.

Pertanyaan Kuis

Apa yang akan dicetak?

```
var signedIn;  
console.log(signedIn);
```

☒ belum diartikan



☐ batal

☐ NaN

☐ kesalahan

Kirim

26. Equality

Dalam kasus perbandingan reguler, operan di kedua sisi operator == pertama-tama diubah menjadi angka, sebelum dibandingkan. Oleh karena itu, a '', false, dan 0 semuanya dianggap sama. Demikian pula, a '1' dan 1 juga dianggap sama. Jika kita tidak ingin mengonversi operan, sebelum melakukan perbandingan, kita harus menggunakan strict comparison ===, yang dijelaskan di bawah ini.

Pertanyaan Kuis

Menurut Anda nilai apa yang "Hello" % 10 akan dihasilkan?

☐ 0

☐ "Halo10"

☐ 10

☐ Kesalahan sintaks

☒ NaN



Kirim

27. Quiz: Semicolons! (2-8)

```
1 /*  
2  * Programming Quiz: Semicolons! (2-8)  
3  */  
4  
5 /*  
6  * QUIZ REQUIREMENTS  
7  * 1. Your code should have two variables - `thingOne` and `thingTwo`  
8  * 2. Your code should only have one `console.log` statement  
9  * 3. Your code should print out the values of `thingOne` and `thingTwo` using concatenation  
10 * 4. Your code should contain semicolons at the end of each line  
11 */  
12 var thingone = 8;  
13 var thingtwo = 10;  
14 console.log(thingone+thingtwo);
```

18

28. Quiz: What's my Name? (2-9)

```
1 /*  
2  * Programming Quiz: What's my Name? (2-9)  
3  */  
4  
5 /*  
6  * QUIZ REQUIREMENTS  
7  * 1. Your code should have a variable `fullName`  
8  * 2. Your variable `fullName` should be declared using the `var` keyword  
9  * 3. Your variable `fullName` should be a non-empty string  
10 */  
11  
12 // your code goes here  
13 let fullName = "John Doe";  
14 console.log(fullName)
```

John Doe

29. Quiz: Out to Dinner (2-10)

```
1- /*
2  * Programming Quiz: Out to Dinner (2-10)
3  */
4- /*
5  * QUIZ REQUIREMENTS
6  * 1. Your code should have the variables - `bill`, `tip`, and `total`
7  * 2. Your variables - `bill`, `tip`, and `total` should be declared using the `var` keyword
8  * 3. Your variable `bill` should be a number, having a value equal to the result of `10.25 + 3.99 + 7.15`
9  * 4. Your variable `tip` should be a number, having a value equal to 15% of the `bill`
10 * 5. Your variable `total` should be a number, having a value equal to the `bill` and `tip` added together
11 * 6. Your code should print the total to the console
12 */
13
14 // your code goes here
15 let bill = 10.25 + 3.99 + 7.15;
16 let tip = bill * 0.15;
17 let total = bill + tip;
18
19 console.log("Total tagihan: $" + total.toFixed(2));
```

Total tagihan: \$24,60

30. Quiz: Mad Libs (2-11)

```
1- /*
2  * Programming Quiz: MadLibs (2-11)
3  */
4- /*
5  * QUIZ REQUIREMENTS
6  * 1. Your code should have the variables - `adjective1`, `adjective2`, `adjective3`, and `madLib`.
7  *
8  * 2. Use the adjective1, adjective2, and adjective3 variables to set the madLib variable to the message:
9  * 'The Intro to JavaScript course is amazing. James and Julia are so fun. I cannot wait to work through
10 *
11 * 3. Your madLib should match the given string
12 * 4. The madLib variable should be printed to the console
13 */
14
15 // your code goes here
16 var adjective1 = "amazing";
17 var adjective2 = "fun";
18 var adjective3 = "entertaining";
19
20 var madLib = "The Intro to JavaScript course is " + adjective1 + ". James and Julia are so " + adjective2
21
22 console.log(madLib);
23
```

The Intro to JavaScript course is amazing. James and Julia are so fun. I cannot wait to work through the rest of this entertaining content!

31. One Awesome Message (2-12)

```
8  * 3. The `awesomeMessage` variable should have the correct format, as shown below:
9  * "Hi, my name is _____. I love _____. In my spare time, I like to _____."
10 *
11 * 4. Print `awesomeMessage` variable to the console using log() method
12 */
13
14 /*
15 * Example:
16 * - Assuming, firstName would have been assigned to "Julia",
17 *   interest to "cats", and hobby to "play video games" to produce the following message:
18 *
19 *   Hi, my name is Julia. I love cats. In my spare time, I like to play video games.
20 *
21 * - Be sure to include spaces and periods where necessary!
22 */
23
24
25 var firstName = "Julia";
26 var interest = "cats";
27 var hobby = "play video games";
28 var awesomeMessage = "Hi, my name is " + firstName + ". I love " + interest + ". In my spare time, I like to " + hobby + ".";
29 console.log(awesomeMessage);
30
```

Hai, nama saya Julia. Saya suka kucing. Di waktu luang saya, saya suka bermain video game.

32. Lesson 2 Summary

Javascript Data Types

- Number
- Strings
- Booleans
- Null undefined

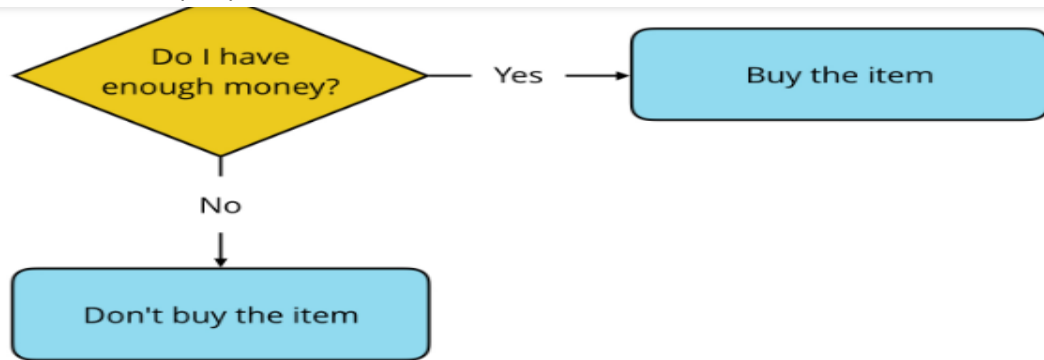
Storing Data in Variables

- Basic Operations

33. Intro to Conditionals

"Conditionals" adalah salah satu konsep dasar dalam pemrograman. Mereka memungkinkan program untuk membuat keputusan berdasarkan kondisi yang diberikan. Dalam JavaScript, terdapat beberapa bentuk conditional, tetapi yang paling umum adalah if statement.

34. Quiz: Flowcharts (3-1)

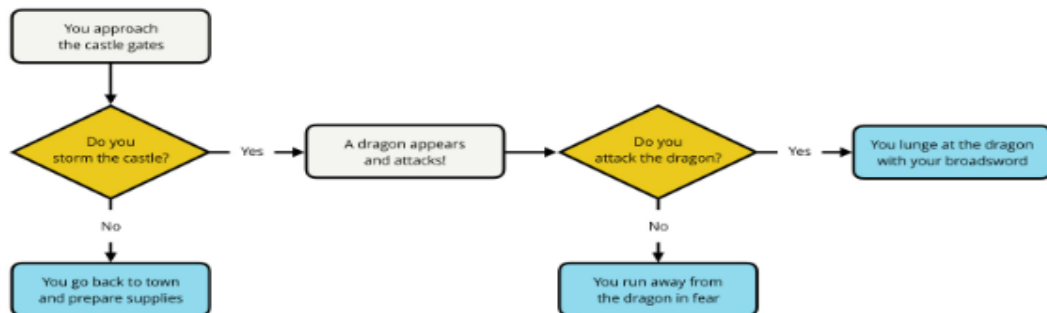


Perhatikan diagram alur di atas. Jenis data apa yang paling baik mewakili (Ya/Tidak) jika Anda memiliki cukup uang untuk membeli barang tersebut?

- ☐ Nomor
- ☐ Rangkaian
- ☒ Boolean
- ☐ belum diartikan



Pertanyaan Kuis



Berikut adalah bagan alur yang berbeda untuk mendekati gerbang kastil. Jika Anda memutuskan untuk menyerbu kastil, apa hasil langsung yang dihasilkan?

- ☒ Seekor naga muncul dan menyerang!
- ☐ Anda kembali ke kota dan menyiapkan persediaan
- ☐ Anda lari dari naga dalam ketakutan
- ☐ Anda menerjang naga dengan pedang Anda



35. Flowchart to Code

Flowchart adalah sebuah diagram yang menggambarkan alur suatu proses atau program. Flowchart dapat digunakan sebagai alat bantu dalam merancang program dan juga dalam memahami alur suatu program.

Untuk mengimplementasikan flowchart ke dalam bentuk kode program, kita dapat mengikuti langkah-langkah berikut:

- Identifikasi kebutuhan program
- Gunakan kondisional
- Gunakan variable
- Gunakan loop
- Implementasikan operasi matematika
- Debugging
- Dokumentasi

36. If...Else Statements

If...else statements adalah jenis kondisional di dalam bahasa pemrograman yang memungkinkan program untuk mengambil keputusan berdasarkan kondisi tertentu. Kondisi yang dievaluasi harus berupa Boolean, yaitu bernilai true atau false. Jika kondisi bernilai true, maka blok kode di dalam if statement akan dieksekusi. Jika kondisi bernilai false, maka blok kode di dalam else statement akan dieksekusi.

37. Else If Statements

Pertanyaan Kuis

Apa yang akan dicetak ke konsol jika kode berikut dijalankan?

```
var money = 100.50;
var price = 100.50;

if (money > price) {
  console.log("You paid extra, here's your change.");
} else if (money === price) {
  console.log("You paid the exact amount, have a nice day!");
} else {
  console.log("That's not enough, you still owe me money.");
}
```

- ☐ "Kamu membayar ekstra, ini kembaliannya."
- ☒ "Kamu membayar jumlah yang tepat, semoga harimu menyenangkan!"
- ☐ "Itu tidak cukup, kamu masih berutang uang padaku."
- ☐ Bukan dari salah satu di atas



Melihat kode berikut, tentukan medali apa yang diterima Kendyll.

```
var runner = "Kendyll";
var position = 2;
var medal;

if(position === 1) {
  medal = "gold";
} else if(position === 2) {
  medal = "silver";
} else if(position === 3) {
  medal = "bronze";
} else {
  medal = "pat on the back";
}

console.log(runner + " received a " + medal + " medal.");
```

☐ Kendyll menerima medali emas.

☒ Kendyll menerima medali perak.



☐ Kendyll menerima medali perunggu.

☐ Kendyll menerima tepukan di punggung.

38. Quiz: Even or Odd (3-2)

```
1- /*
2  * Programming Quiz: Even or Odd (3-2)
3  *
4  * Write an if...else statement that prints `even` if the
5  * number is even and prints `odd` if the number is odd.
6  *
7  * Note - make sure to print only the string "even" or the string "odd"
8  */
9
10- /*
11  * QUIZ REQUIREMENTS
12  * 1. Your code should have a variable `number`, and include an `if...else` statement
13  * 2. Your conditional should use a strict comparison (`===` or `!==`), and use the modulo (`%`) operator
14  * 3. Your code should produce the expected output - "even" or "odd" using console.log() method
15  */
16
17
18 var number = 21;
19
20 // Note that a strict equality operator === checks the equality of operands without type conversion.
21 if (0 === number % 2) {
22   console.log("even");
23 } else {
24   console.log("odd");
25 }
26
```

aneh

39. Quiz: Musical Groups (3-3)

```
1- /*
2-  * Programming Quiz: Musical Groups (3-3)
3-  */
4- /*
5-  * QUIZ REQUIREMENTS
6-  * 1. Your code should have a variable `musicians`, and include `if...else if...else` conditional statement
7-  * 2. Your code should produce the expected output, as mentioned above. Read each condition carefully.
8-  */
9
10 // change the value of `musicians` to test your conditional statements
11 var musicians = 0;
12
13 // your code goes here
14 if (musicians <= 0) {
15   console.log("not a group");
16 } else if (musicians === 1) {
17   console.log("solo");
18 } else if (musicians === 2) {
19   console.log("duet");
20 } else if (musicians === 3) {
21   console.log("trio");
22 } else if (musicians === 4) {
23   console.log("quartet");
24 } else if (musicians >= 5) {
25   console.log("this is a large group");
26 }
27
```

bukan grup

40. Quiz: Murder Mystery (3-4)

```
1 // Assigning values to variables
2 var room = "dining room";
3 var suspect = "Mr. Parkes";
4 var weapon;
5
6 // Conditional statements to determine weapon based on room
7 if (room === "ballroom") {
8   weapon = "poison";
9 } else if (room === "gallery") {
10  weapon = "trophy";
11 } else if (room === "billiard room") {
12  weapon = "pool stick";
13 } else if (room === "dining room") {
14  weapon = "knife";
15 }
16
17 // Conditional statement to determine if suspect was in the room
18 var solved = false;
19
20 if (suspect === "Mr. Parkes" && room === "dining room") {
21   solved = true;
22 } else if (suspect === "Ms. Van Cleve" && room === "gallery") {
23   solved = true;
24 } else if (suspect === "Mrs. Sparr" && room === "billiard room") {
25   solved = true;
26 } else if (suspect === "Mr. Kalehoff" && room === "ballroom") {
27   solved = true;
28 }
29
```

Tuan Parkes melakukannya di ruang makan dengan pisau!

41. More Complex Problem

Masalah yang lebih kompleks di bidang JavaScript bisa mencakup beberapa area yang berbeda, termasuk:

- Aplikasi web yang kompleks: Aplikasi web modern yang kompleks memerlukan banyak kode JavaScript untuk menjalankan fungsionalitas yang berbeda, seperti manipulasi DOM, membuat animasi, pengiriman permintaan jaringan, dan banyak lagi.
- Optimisasi kinerja: Dalam aplikasi web yang besar, kinerja bisa menjadi masalah besar. Ini melibatkan optimisasi kode JavaScript untuk meningkatkan waktu respon dan meminimalkan waktu muat, serta menerapkan teknik seperti caching dan kompresi.
- Pengembangan lintas platform: Masalah ini muncul ketika Anda ingin membuat aplikasi yang dapat berjalan pada banyak platform, seperti desktop, web, dan seluler. Ini memerlukan pengetahuan tentang berbagai teknologi, seperti React Native, Angular, dan Cordova.
- Debugging masalah: Dalam kode JavaScript yang kompleks, debugging dapat menjadi sulit karena kompleksitas kode. Hal ini memerlukan keterampilan dan alat untuk membantu memecahkan masalah.
- Integrasi dengan layanan pihak ketiga: Integrasi dengan layanan pihak ketiga seperti API, database, dan platform lainnya, bisa menjadi masalah yang kompleks dan memerlukan pemahaman tentang protokol dan format data yang berbeda.
- Keamanan: Masalah keamanan bisa sangat kompleks di aplikasi web modern. Hal ini melibatkan melindungi data pengguna, mencegah serangan jaringan, dan menjaga kode JavaScript tetap aman dan terlindungi.

42. Logical Operators

Simbolnya && adalah operator logika AND, dan digunakan untuk menggabungkan dua ekspresi logika menjadi satu ekspresi logika yang lebih besar. Jika kedua ekspresi yang lebih kecil adalah benar, maka seluruh ekspresi dievaluasi menjadi benar. Jika salah satu dari ekspresi yang lebih kecil adalah false, maka seluruh ekspresi logis adalah false.

Pertanyaan Kuis

Berapa nilai [BLANK] yang akan membuat ekspresi berikut bernilai `false`. Perhatikan yang `!` benar di awal!

`!([BLANK] === 4) && "STRing" === "STRing"`

Pengembalian: salah

☐ -4

☒ 4

☐ "4"

☐ "-4"

Pertanyaan Kuis

Pilih operator yang akan membuat ekspresi berikut menjadi `true`.

`3 < -10 [BLANK] "James" !== "james"`

Pengembalian: benar

☐ `!`

☒ `||`

☐ `&&`

Pertanyaan Kuis

Evaluasi ekspresi logis berikut. Periksa yang mengevaluasi ke `true`.

☒ `true || false`

☐ `false && false`

☐ `!true`

☒ `(13 > -7) || (false == 0)`

☐ `(10 === "10") && (1 <= 2)`

☒ `(3 != 6 % 3) && !(24 > 45) && (!false)`

43. Logical AND and OR

Logika AND (&&) dan logika OR (||) adalah operator logika pada bahasa pemrograman yang digunakan untuk mengkombinasikan kondisi dalam suatu ekspresi boolean.

Operator logika AND (&&) digunakan untuk memeriksa apakah semua operand dalam ekspresi boolean bernilai true. Jika salah satu atau semua operand bernilai false, maka ekspresi akan menghasilkan nilai false. Namun jika semua operand bernilai true, maka ekspresi akan menghasilkan nilai true. Contohnya, ekspresi `5 > 3 && 2 < 4` akan menghasilkan nilai true karena kedua operand, yaitu `5 > 3` dan `2 < 4`, bernilai true.

Operator logika OR (||) digunakan untuk memeriksa apakah setidaknya satu operand dalam ekspresi boolean bernilai true. Jika semua operand bernilai false, maka ekspresi akan menghasilkan nilai false. Namun jika setidaknya satu operand bernilai true, maka ekspresi akan menghasilkan nilai true. Contohnya, ekspresi `5 < 3 || 2 > 4` akan menghasilkan nilai false karena kedua operand, yaitu `5 < 3` dan `2 > 4`, bernilai false.

44. Quiz: Checking your Balance (3-5)

```
2 var balance = 325.00;
3 var checkBalance = true;
4 var isActive = false;
5
6 // your code goes here
7 // change the values of `balance`, `checkBalance`, and `isActive` to test your code
8 var balance = -325;
9 var checkBalance = true;
10 var isActive = true;
11
12 // Start only if checkBalance === true
13 if (checkBalance === true) {
14   // Cases when account is active. Now, the balance could be <, =, or > zero
15   if (isActive === true && balance > 0) {
16     console.log("Your balance is $" + balance.toFixed(2) + ".");
17   }
18   else if (isActive === true && balance === 0) {
19     console.log("Your account is empty.");
20   }
21   else if (isActive === true && balance < 0) {
22     console.log("Your balance is negative. Please contact bank.");
23   }
24   // Case when account is NOT active
25   else if (isActive === false) {
26     console.log("Your account is no longer active.");
27   }
28 }
29 else {
30   console.log("Thank you. Have a nice day!");
31 }
```

Saldo Anda negatif. Silakan hubungi bank.

45. Quiz: Ice Cream (3-6)

```
1 * - `flavor=chocolate`, `vessel=cone`, and `toppings=sprinkles`
2 * - `flavor=chocolate`, `vessel=cone`, and `toppings=peanuts`
3 * - `flavor=chocolate`, `vessel=bowl`, and `toppings=sprinkles`
4 * - `flavor=chocolate`, `vessel=bowl`, and `toppings=peanuts`
5 *
6 * 5. Your code should NOT log (print) anything when
7 * - the flavor is something other than "vanilla" or "chocolate"
8 * - the vessel is something other than "cone" or "bowl"
9 * - the toppings is something other than "sprinkles" or "peanuts"
10 *
11 * 6. Your code should not be empty
12 */
13
14 // change the values of `flavor`, `vessel`, and `toppings` to test your code
15 var flavor = "strawberry";
16 var vessel = "cone";
17 var toppings = "cookies";
18
19 // Add your code here
20
21 // change the values of `flavor`, `vessel`, and `toppings` to test your code
22 var flavor = "strawberry";
23 var vessel = "cone";
24 var toppings = "cookies";
25
26 // Add your code here
27 if ((flavor === "vanilla" || flavor === "chocolate") && (vessel === "cone" || vessel === "bowl") && (toppings === "sprinkles" || toppings === "peanuts")) {
28   console.log("I'd like two scoops of " + flavor + " ice cream in a " + vessel + " with " + toppings + ".");
29 }
```

46. Quiz: What do I Wear? (3-7)

Kode Anda:

```
1 // change the values of `shirtwidth`, `shirtlength`, and `shirtsleeve` to test your code
2 var shirtwidth = 21;
3 var shirtlength = 99;
4 var shirtsleeve = 8.40;
5
6 var size = "NA";
7
8 if ((shirtwidth>=18 && shirtwidth<20) && (shirtlength>=28 && shirtlength<29) && (shirtsleeve>=8.13 && shirtsleeve<8.38) ) {
9   size = "S";
10 }
11 else if ((shirtwidth>=20 && shirtwidth<22) && (shirtlength>=29 && shirtlength<30) && (shirtsleeve>=8.38 && shirtsleeve<8.63) ) {
12   size = "M";
13 }
14 else if ((shirtwidth>=22 && shirtwidth<24) && (shirtlength>=30 && shirtlength<31) && (shirtsleeve>=8.63 && shirtsleeve<8.88) ) {
15   size = "L";
16 }
17 else if ((shirtwidth>=24 && shirtwidth<26) && (shirtlength>=31 && shirtlength<33) && (shirtsleeve>=8.88 && shirtsleeve<9.63) ) {
18   size = "XL";
19 }
20 else if ((shirtwidth>=26 && shirtwidth<28) && (shirtlength>=33 && shirtlength<34) && (shirtsleeve>=9.63 && shirtsleeve<10.13) ) {
21   size = "2XL";
22 }
23 else if ((shirtwidth>=28) && (shirtlength>=34) && (shirtsleeve>=10.13) ) {
24   size = "3XL";
25 }
26 else {
27   size = "NA";
28 }
29 console.log(size);
```

NA

47. Advanced Conditionals

Kondisi lanjutan pada bahasa pemrograman digunakan untuk mengeksekusi suatu blok kode jika suatu kondisi tertentu terpenuhi, dan mengeksekusi blok kode yang berbeda jika kondisi tersebut tidak terpenuhi.

48. Truthy and Falsy

Truthy dan Falsy adalah istilah dalam JavaScript untuk menggambarkan nilai-nilai yang dianggap sebagai true atau false dalam konteks boolean.

Quiz Question
Select the truthy values from the list of values.

<input type="checkbox"/>	<code>""</code>	
<input checked="" type="checkbox"/>	<code>"null"</code>	✓
<input checked="" type="checkbox"/>	<code>-5</code>	✓
<input type="checkbox"/>	<code>0.0</code>	
<input type="checkbox"/>	<code>undefined</code>	

49. Ternary Operator

Untuk menggunakan operator ternary, pertama berikan pernyataan kondisional di sisi kiri ?. Kemudian, di antara ? dan : tulis kode yang akan dijalankan jika kondisinya adalah true dan di sisi kanan kode tuliskan :kode yang akan dijalankan jika kondisinya false.

Pertanyaan Kuis
Apa yang akan dicetak ke konsol jika kode berikut dijalankan?

```
var adult = true;
var preorder = true;

console.log("It costs $" + (adult ? "40.00" : "20.00") + " to attend the concert. Pick up your tickets at the " + (preorder ? "will call" : "gate") + ".");
```

<input type="radio"/>	Biayanya \$20,00 untuk menghadiri konser. Ambil tiket Anda di pintu gerbang.	
<input type="radio"/>	Biayanya \$20,00 untuk menghadiri konser. Ambil tiket Anda di telepon wasiat.	
<input type="radio"/>	Biayanya \$40,00 untuk menghadiri konser. Ambil tiket Anda di pintu gerbang.	
<input checked="" type="radio"/>	Biayanya \$40,00 untuk menghadiri konser. Ambil tiket Anda di telepon wasiat.	✓

50. Quiz: Navigating the Food Chain (3-8)

```
1  * - use the variables `eatsPlants` and `eatsAnimals` in your ternary expressions
2  * - `if` statements aren't allowed ;- )
3  */
4
5  /*
6  * QUIZ REQUIREMENTS
7  * - Your code should have the variables `eatsPlants`, `eatsAnimals`
8  * - Your code should include ternary statements. Do not use if....else statement.
9  * - Your code should produce the expected output
10 * - Your code should not be empty
11 * - BE CAREFUL ABOUT THE PUNCTUATION AND THE EXACT WORDS TO BE PRINTED.
12 */
13
14 // change the values of `eatsPlants` and `eatsAnimals` to test your code
15 var eatsPlants = false;
16 var eatsAnimals = true;
17
18 /*
19 * Test your code against the following possible input/output combinations of (`eatsPlants`, `eatsAnimals`, expected output):
20 * - (true, true, omnivore)
21 * - (false, true, carnivore)
22 * - (true, false, herbivore)
23 * - (false, false, undefined)
24 */
25
26 var category = eatsPlants ? (eatsAnimals ? "omnivore" : "herbivore") : (eatsAnimals ? "carnivore" : "undefined");
27
28 console.log(category);
29
```

karnivora

51. Switch Statement

Switch Statement adalah salah satu struktur kontrol pada bahasa pemrograman JavaScript yang digunakan untuk melakukan seleksi nilai atau ekspresi tertentu. Struktur kontrol ini akan mengevaluasi nilai atau ekspresi yang diberikan dan membandingkannya dengan beberapa nilai atau ekspresi lain yang sudah ditentukan sebelumnya.

```
switch(month) {
  case 1:
  case 3:
  case 5:
  case 7:
  case 8:
  case 10:
  case 12:
    days = 31;
    break;
  case 4:
  case 6:
  case 9:
  case 11:
    days = 30;
    break;
  case 2:
    days = 28;
}

console.log("There are " + days + " days in this month.");
```

☐ Ada 31 hari dalam bulan ini.

☐ Ada 30 hari dalam bulan ini.

☒ Ada 28 hari dalam bulan ini.



52. Falling-through

Falling-through pada JavaScript terjadi ketika kasus (case) dalam blok switch statement tidak memiliki instruksi break atau return, sehingga kontrol aliran program akan terus berlanjut ke kasus selanjutnya tanpa menghentikan eksekusi

Pertanyaan Kuis

Jika `winner` sama dengan 3, lalu apa yang akan ditampilkan ke konsol?

```
var prize = "";  
  
switch (winner) {  
  case 1:  
    prize += "a trip for two to the Bahamas and ";  
  case 2:  
    prize += "a four piece furniture set.";  
    break;  
  case 3:  
    prize += "a smartwatch and ";  
  default:  
    prize += "tickets to the circus.";  
}  
  
console.log("You've won " + prize);
```

☒ Anda memenangkan jam tangan pintar dan tiket ke sirkus. ✓

☐ Anda telah memenangkan tiket ke sirkus.

☐ Anda telah memenangkan satu set furnitur empat bagian.

☐ Anda telah memenangkan perjalanan untuk dua orang ke Bahama dan satu set furnitur empat bagian.

53. Lesson 3 Summary

Break Down Problems

Conditional Statements And Logical Operators

Advanced Techniques

- Truthy Falsy
- Ternary Operator
- Switch Statements

54. Intro to Loops

Loop pada JavaScript adalah struktur kontrol yang digunakan untuk mengulang eksekusi blok kode tertentu berulang kali selama kondisi tertentu masih terpenuhi.

55. While Loops

While Loops pada JavaScript adalah jenis loop yang digunakan untuk mengulang eksekusi blok kode tertentu selama kondisi tertentu masih terpenuhi. Loop ini akan terus berjalan selama kondisi yang diberikan pada bagian while masih bernilai true.

56. Parts of a While Loop

Tiga informasi utama yang harus dimiliki setiap loop adalah:

- Kapan memulai: Kode yang mengatur perulangan — misalnya menentukan nilai awal variabel.
- Kapan harus berhenti: Kondisi logis untuk menguji apakah perulangan harus dilanjutkan.
- Cara mendapatkan item berikutnya: Langkah penambahan atau pengurangan — misalnya, $x = x * 3$ atau $x = x - 1$

Pertanyaan Kuis

Berapa kali while loop akan berjalan?

```
var x = 10;
while (x <= 25) {
  console.log('Printing out x = ' + x);
  x = x + 2;
}
```

☐ 15

☐ 16

☒ 8



☐ 9

Pertanyaan Kuis

Inilah while loop yang seharusnya mencetak nilai `x` dari 0 sampai 5, tapi ada bug. Apa yang hilang?

```
while (x < 6) {
  console.log('Printing out x = ' + x);
}
```

☒ x tidak pernah bertambah



☐ Tidak ada syarat untuk memberi tahu loop kapan harus berhenti

☒ x tidak pernah diberi nilai awal



☐ Pernyataan if digunakan sebagai pengganti while loop

57. Quiz: JuliaJames (4-1)

```
1  * QUIZ REQUIREMENTS
2  * - Your code should have a variable `x` with a starting value of `1`
3  * - Your code should include a `while` loop
4  * - Your `while` loop should have a stop condition
5  * - Your code should use a conditional statement
6  * - Your code should increment `x` by `1` each time the loop executes
7  * - Your code should produce the expected output
8  * - Your code should not be empty
9  * - BE CAREFUL ABOUT THE PUNCTUATION AND THE EXACT WORDS TO BE PRINTED.
10 */
11
12 var x = 1;
13
14 // while loop with a stop condition
15 while (x <= 20){
16     if (x % 3 === 0 && x % 5 === 0) {
17         console.log("JuliaJames");
18     }else if (x % 5 === 0) {
19         console.log("James");
20     }else if (x % 3 === 0) {
21         console.log("Julia");
22     }
23     else {
24         console.log(x);
25     }
26     // increment x at the end of each loop
27     x = x + 1 ;
28 }
29
30
```

```
1 2 Julia 4 James Julia 7 8 Julia James 11 Julia 13 14 JuliaJames 16 17 Julia 19 James
```

58. Quiz: 99 Bottles of Juice (4-2)

```
5
6
7 var num = 99;
8
9 while (num >= 1){
10     // Last iteration. Note occurrences of bottle, bottle, bottles
11     if (num === 1) {
12         console.log(num + " bottle of juice on the wall! "
13             + num + " bottle of juice! Take one down, pass it around... "
14             + (num-1) + " bottles of juice on the wall!");
15     }
16     // Second-last iteration. Note occurrences of bottles, bottles, bottle
17     else if (num === 2){
18         console.log(num + " bottles of juice on the wall! "
19             + num + " bottles of juice! Take one down, pass it around... "
20             + (num-1) + " bottle of juice on the wall!");
21     }
22     // All other iterations. Note occurrences of bottles, bottles, bottles
23     else {
24         console.log(num + " bottles of juice on the wall! "
25             + num + " bottles of juice! Take one down, pass it around... "
26             + (num-1) + " bottles of juice on the wall!");
27     }
28     num = num - 1;
29 }
30
```

```
99 botol jus di dinding! 99 botol jus! Ambil satu, sebarakan... 98 botol jus di dinding! 98 botol jus di
dinding! 98 botol jus! Ambil satu, sebarakan... 97 botol jus di dinding! 97 botol jus di dinding! 97
botol jus! Ambil satu, sebarakan... 96 botol jus di dinding! 96 botol jus di dinding! 96 botol jus!
Ambil satu, bagikan sekitar... 95 botol jus di dinding! 95 botol jus di dinding! 95 botol jus! Ambil
satu, sebarakan... 94 botol jus di dinding! 94 botol jus di dinding! 94 botol jus! Ambil satu,
sebarakan... 93 botol jus di dinding! 93 botol jus di dinding! 93 botol jus! Ambil satu, sebarakan... 92
botol jus di dinding! 92 botol jus di dinding! 92 botol jus! Ambil satu, sebarakan... 91 botol jus di
dinding! 91 botol jus di dinding! 91 botol jus! Ambil satu, bagikan sekitar... 90 botol jus di dinding!
90 botol jus di dinding! 90 botol jus! Ambil satu, sebarakan... 89 botol jus di dinding! 89 botol jus di
dinding! 89 botol jus! Ambil satu, sebarakan... 88 botol jus di dinding! 88 botol jus di dinding! 88
botol jus! Ambil satu, sebarakan... 87 botol jus di dinding! 87 botol jus di dinding! 87 botol jus!
Ambil satu, sebarakan... 86 botol jus di dinding! 86 botol jus di dinding! 86 botol jus! Ambil satu,
```

59. Quiz: Countdown, Liftoff! (4-3)

```
2- while (n >= 0){
3-   if (n === 50) {
4-     console.log("Orbiter transfers from ground to internal power");
5-   }
6-   else if (n === 31) {
7-     console.log("Ground launch sequencer is go for auto sequence start");
8-   }
9-   else if (n === 16) {
10-    console.log("Activate launch pad sound suppression system");
11-  }
12-  else if (n === 10) {
13-    console.log("Activate main engine hydrogen burnoff system");
14-  }
15-  else if (n === 6) {
16-    console.log("Main engine start");
17-  }
18-  else if (n === 0) {
19-    console.log("Solid rocket booster ignition and liftoff!");
20-  }
21-  else{
22-    console.log("T-"+n+" seconds");
23-  }
24-  //Never forget to decrement/increment the iteration variable in a while loop
25-  // Otherwise, you loop will run infinite iterations
26-  n = n-1;
27-}
28-
```

T-60 detik T-59 detik T-58 detik T-57 detik T-56 detik T-55 detik T-54 detik T-54 detik T-53 detik T-52 detik T-51 detik Pengorbit mentransfer dari ground ke daya internal T-49 detik T-48 detik T-47 detik T-46 detik T-45 detik T-44 detik T-43 detik T-42 detik T-41 detik T-40 detik T-39 detik T-38 detik T-37 detik T-36 detik T-35 detik T-34 detik T-33 detik T-32 detik Ground launch sequencer digunakan untuk mulai urutan otomatis T-30 detik T-29 detik T-28 detik T-27 detik T-26 detik T-25 detik T-24 detik T-23 detik T-22 detik T-21 detik T-20 detik T-19 detik T-18 detik T-17 detik Aktifkan sistem peredam suara landasan peluncuran T-15 detik T-14 detik T-13 detik T-12 detik T-11 detik Mengaktifkan sistem pembakaran hidrogen mesin utama T-9 detik T-8 detik T-7 detik Mesin utamastart T-5 detik T-4 detik T-3 detik T-2 detik T-1 detik Penyalaan dan peluncuran solid rocket booster!

60. For Loops

For loop pada JavaScript digunakan untuk mengulang sebuah blok kode tertentu sejumlah kali. For loop terdiri dari tiga bagian utama yaitu inisialisasi variabel, kondisi looping, dan perubahan variabel setelah setiap iterasi.

61. Parts of a For Loop

Berikut adalah contoh perulangan for yang mencetak nilai dari 0 hingga 5. Perhatikan titik koma yang memisahkan berbagai pernyataan perulangan for: var i = 0; i < 6; i = i + 1

```
for (var i = 0; i < 6; i = i + 1) {
  console.log("Printing out i = " + i);
}
```

Cetakan :

Mencetak i = 0
Mencetak i = 1
Mencetak i = 2
Mencetak i = 3
Mencetak i = 4
Mencetak i = 5

62. Nested Loops

Nested loops pada JavaScript adalah sebuah konstruksi perulangan yang terdiri dari satu atau lebih loop di dalam loop lainnya. Dalam nested loops, setiap perulangan akan mengeksekusi seluruh iterasi yang ada di dalam perulangan lainnya.

Quiz Question
What will this loop print out?

```
for (var i = 0; i <= 6; i = i + 2) {  
  console.log(i);  
}
```

☐ 1 2 3 4 5 6

☐ 0 1 2 3 4 5 6

☐ 1 3 5

☐ 1 3 5 7

☐ 0 2 4

☒ 0 2 4 6

63. Increment and Decrement

Increment dan Decrement adalah operator yang digunakan untuk menambah atau mengurangi nilai dari sebuah variabel dengan jumlah 1. Dalam JavaScript, terdapat dua jenis operator Increment dan Decrement, yaitu:

++ (Increment): operator ini digunakan untuk menambahkan nilai variabel dengan jumlah 1. Contohnya:

```
var x = 5;  
x++; // sama dengan x = x + 1;  
console.log(x); // output: 6
```

-- (Decrement): operator ini digunakan untuk mengurangi nilai variabel dengan jumlah 1. Contohnya:

```
var y = 10;  
y--; // sama dengan y = y - 1;  
console.log(y); // output: 9
```

64. Quiz: Changing the Loop (4-4)

```
1  /*
2   * Programming Quiz: Changing the Loop (4-4)
3   */
4  /*
5   * QUIZ REQUIREMENTS
6   * - Your code should use a `for` loop
7   * - Your `for` loop should specify a starting condition `x` with the value of `9`
8   * - Your `for` loop should have a stop condition for the value of `x`
9   * - Your `for` loop should decrement `x` each time it executes
10  * - Your code should produce the expected output, as explained above
11  * - Your code should not be empty
12  * - BE CAREFUL ABOUT THE PUNCTUATION AND THE EXACT WORDS TO BE PRINTED.
13  */
14
15  for (var x = 9; x>=1; x--){
16      console.log("hello " + x);
17  }
```

```
hello 9 hello 8 hello 7 hello 6 hello 5 hello 4 hello 3 hello 2 hello 1
```

65. Quiz: Fix the Error 1 (4-5)

```
4
5  /*
6   * QUIZ REQUIREMENTS
7   * - Your code should use a `for` loop
8   * - Your `for` loop should specify a starting condition `x` with the value of `5`
9   * - Your `for` loop should have a stop condition for the value of `x`
10  * - Your `for` loop should increment `x` each time it executes
11  * - Your code should produce the expected output, as explained above
12  * - Your code should not be empty
13  * - BE CAREFUL ABOUT THE PUNCTUATION AND THE EXACT WORDS TO BE PRINTED.
14  */
15
16  for (var x = 5; x < 10; x++) {
17      console.log(x);
18  }
```

```
5 6 7 8 9
```


66. Quiz: Fix the Error 2 (4-6)

```
2  * PROGRAMMING QUIZ: FIX THE ERROR 2 (4-6)
3  */
4  /*
5  * QUIZ REQUIREMENTS
6  * - Your code should use a `for` loop
7  * - Your `for` loop should specify a starting condition `k` with the value of `0`
8  * - Your `for` loop should have a stop condition for the value of `k`
9  * - Your `for` loop should increment `k` each time it executes
10 * - Your code should produce the expected output, as explained above
11 * - Your code should not be empty
12 * - BE CAREFUL ABOUT THE PUNCTUATION AND THE EXACT WORDS TO BE PRINTED.
13 */
14
15 // The semicolon was missing at two places
16 for (var k = 0; k < 200; k++) {
17     console.log(k);
18 }
19
```

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131
132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157
158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183
184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199
```

67. Quiz: Factorials! (4-7)

```
1  /*
2  * Programming Quiz: Factorials (4-7)
3  */
4
5  // your code goes here
6
7  var solution = 1;
8  for (var i = 1; i <= 12; i++) {
9      solution *= i;
10 }
11 console.log(solution);
12
```

479001600

68. Quiz: Find my Seat (4-8)

```
17  */
18
19  // Write your code here
20  var row = 0; // initial value of the row
21  var seat = 0; // initial value of the seat within a row
22
23  // One loop inside another is called Nested loop.
24  // Outer `for` loop, to iterate over the rows
25  for (row = 0; row <= 25; row++){
26
27      // Inner `for` loop, to iterate over the seats within a row
28      // In this loop, the value of `row` variable would change only after 100 iterations
29      for(seat = 0; seat <= 99; seat++){
30          console.log(row+"-"+seat);
31      }
32  }
```

```
0-0 0-1 0-2 0-3 0-4 0-5 0-6 0-7 0-8 0-9 0-10 0-11 0-12 0-13 0-14 0-15 0-16 0-17 0-18 0-19 0-20 0-21 0-22 0-23 0-24 0-25 0-26 0-27 0-28 0-29 0-30 0-31 0-32 0-33 0-34 0-35 0-36 0-37 0-38 0-39 0-40 0-41 0-42 0-43 0-44 0-45 0-46 0-47 0-48 0-49 0-50 0-51 0-52 0-53 0-54 0-55 0-56 0-57 0-58 0-59 0-60 0-61 0-62 0-63 0-64 0-65 0-66 0-67 0-68 0-69 0-70 0-71 0-72 0-73 0-74 0-75 0-76 0-77 0-78 0-79 0-80 0-81 0-82 0-83 0-84 0-85 0-86 0-87 0-88 0-89 0-90 0-91 0-92 0-93 0-94 0-95 0-96 0-97 0-98 0-99 1-0 1-1 1-2 1-3 1-4 1-5 1-6 1-7 1-8 1-9 1-10 1-11 1-12 1-13 1-14 1-15 1-16 1-17 1-18 1-19 1-20 1-21 1-22 1-23 1-24 1-25 1-26 1-27 1-28 1-29 1-30 1-31 1-32 1-33 1-34 1-35 1-36 1-37 1-38 1-39 1-40 1-41 1-42 1-43 1-44 1-45 1-46 1-47 1-48 1-49 1-50 1-51 1-52 1-53 1-54 1-55 1-56 1-57 1-58 1-59 1-60 1-61 1-62 1-63 1-64 1-65 1-66 1-67 1-68 1-69 1-70 1-71 1-72 1-73 1-74 1-75 1-76 1-77 1-78 1-79 1-80 1-81 1-82 1-83 1-84 1-85 1-86 1-87 1-88 1-89 1-90 1-91 1-92 1-93 1-94 1-95 1-96 1-97 1-98 1-99 2-0 2-1 2-2 2-3 2-4 2-5 2-6 2-7 2-8 2-9 2-10 2-11 2-12 2-13 2-14 2-15 2-16 2-17 2-18 2-19 2-20 2-21 2-22 2-23 2-24 2-25 2-26 2-27 2-28 2-29 2-30 2-31 2-32 2-33 2-34 2-35 2-36 2-37 2-38 2-39 2-40 2-41 2-42 2-43 2-44 2-45 2-46 2-47 2-48 2-49 2-50 2-51 2-
```

69. Lesson 4 Summary

Loop adalah topik mendasar untuk Bahasa pemrograman apa pun, pastikan anda merasa nyaman dengan loop karena anda akan menggunakan setiap saat. Loop adalah fitur yang sangat keren dari Bahasa tersebut, tetapi anda harus tahu apa yang mungkin menjadi fungsi yang lebih baik , fungsi sangat keren karena membiarkan anda mengelompokkan baris kode bersama sama dan menjalankan semua.

70. Intro to Functions

Function adalah blok kode yang dapat digunakan kembali dan dijalankan ketika dibutuhkan. Dalam JavaScript, function adalah objek yang bisa dianggap sebagai "paket" yang terdiri dari beberapa pernyataan yang akan dieksekusi bersama-sama. Function biasanya digunakan untuk menyelesaikan suatu tugas tertentu, dan dapat menerima input (parameter) dan mengembalikan output.

71. Function Example

```
“Hi, semaJ! My name is ailuJ”

1  function reverseString(reverseMe) {
2      var reversed = "";
3      for (var i = reverseMe.length - 1; i >= 0; i--) {
4          reversed += reverseMe[i];
5      }
6      return reversed;
7  }
8
9  console.log("Hi, " + reverseString("James") + "! My name
10 is " + reverseString("Julia"));
```

72. Declaring Functions

Anda dapat mendeklarasikan fungsi di JavaScript menggunakan function statement atau function expression.

- Function statement dideklarasikan dengan menggunakan kata kunci function, diikuti dengan nama fungsi, daftar parameter dalam tanda kurung, dan kode yang akan dieksekusi di dalam kurung kurawal. Contoh deklarasi fungsi menggunakan function statement:

```
function namaFungsi(param1, param2) {  
    // kode yang akan dieksekusi  
}
```

- Function expression adalah cara lain untuk mendeklarasikan fungsi di JavaScript. Dalam hal ini, fungsi dideklarasikan sebagai nilai variabel. Contoh deklarasi fungsi menggunakan function expression:

```
var namaFungsi = function(param1, param2) {  
    // kode yang akan dieksekusi  
};
```

Pertanyaan Kuis

Gunakan fungsi berikut untuk menjawab pertanyaan ini.

```
function findAverage(x, y) {  
    var answer = (x + y) / 2;  
    return answer;  
}
```

```
var avg = findAverage(5, 9);
```

Nilai apa yang akan disimpan dalam variabel `avg`?

- ☐ "menjawab"
- ☐ $(x + y) / 2$
- ☒ 7
- ☐ 14
- ☐ 4



Pertanyaan Kuis

```
function findAverage(x, y) {  
    var answer = (x + y) / 2;  
    return answer;  
}
```

```
var avg = findAverage(5, 9);
```

Apakah `x` dan `y` parameter atau argumen untuk fungsi ini?

- ☒ Parameter
- ☐ Argumen



Kirim

73. Function Recap

Fungsi mengemas kode sehingga Anda dapat dengan mudah menggunakan (dan menggunakan kembali) blok kode. Parameter adalah variabel yang digunakan untuk menyimpan data yang diteruskan ke fungsi agar fungsi tersebut dapat digunakan. Argumen adalah data aktual yang diteruskan ke suatu fungsi saat dipanggil

74. Quiz: Laugh it Off 1 (5-1)

```
10 // BE CAREFUL ABOUT THE FUNCTIONION AND THE EXACT WORDS TO BE PRINTED.  
11 */  
12  
13  
14 // your code goes here  
15  
16 // function definition  
17 function laugh() {  
18     // Take a variable with any name of your choice  
19     var sound = "hahahahahahahahaha!"  
20     return sound;  
21 }  
22  
23  
24 // function calling  
25 // Don't forget to put a pair of `( )` along with the function name, while calling  
26 console.log(laugh());  
27
```

hahahahahahahahaha!

75. Quiz: Laugh it Off 2 (5-2)

```
17 var sound = "" ;  
18 function laugh(num) {  
19     for (var x = 0 ; x < num ; x++) {  
20         sound = sound + "ha" ;  
21     }  
22     sound = sound + "!";  
23     return sound;  
24 }  
25  
26 console.log(laugh(3)) ;
```

hahaha!

76. Return Values

Dalam JavaScript, sebuah fungsi dapat mengembalikan nilai dengan menggunakan kata kunci `return`. Ketika sebuah fungsi dipanggil, kode di dalam fungsi akan dijalankan dan menghasilkan nilai yang akan dikembalikan ke tempat pemanggilan fungsi.

Pertanyaan Kuis

Apa fungsi ini "kembali"?

```
function sleep() {  
  console.log("I'm sleepy!");  
  return "zzz";  
  return "snore";  
}  
  
sleep();
```

☐ "Saya mengantuk!"

☒ "zzz"



☐ "mendengkur"

Pertanyaan Kuis

Nomor berapa yang akan "dicetak" (ke konsol JavaScript)?

```
function square(x) {  
  return x * x;  
}  
  
function subtractFour(x) {  
  return square(x) - 4;  
}  
  
console.log(subtractFour(5));
```

☐ 25

☐ 1

☒ 21



☐ 5

Pertanyaan Kuis

Menurut Anda apa yang akan terjadi dengan kode berikut?

```
function test() {  
  return 1;  
  return 2;  
}  
  
test();
```

☒ 1 akan dikembalikan



☐ 2 akan dikembalikan

☐ 3 akan dikembalikan

☐ kesalahan

77. Using Return Value

Setelah sebuah fungsi mengembalikan nilai, nilai tersebut dapat digunakan oleh kode lain di program. Misalnya, nilai yang dikembalikan oleh fungsi dapat disimpan dalam variabel atau digunakan sebagai argumen untuk fungsi lain.

Coba prediksi apa yang akan dicetak pada `console.log` pernyataan di bawah ini. Kemudian, periksa prediksi Anda dengan menempelkan kode ke konsol JavaScript. Fungsi bisa jadi rumit, jadi coba cari tahu *sebelum* menjalankan kode!

```
function addTen(x) {  
  return x + 10;  
}  
  
function divideByThree(y) {  
  return y / 3;  
}  
  
var result = addTen(2);  
console.log(divideByThree(result));
```

☐ 2

☐ 10

☐ 12

☒ 4 ✓

78. Coverage

Coverage atau istilah lengkapnya adalah code coverage adalah metrik yang digunakan untuk mengukur sejauh mana kode dalam suatu program telah dieksekusi selama proses pengujian (testing). Code coverage dapat digunakan untuk mengevaluasi kualitas pengujian suatu program, karena semakin banyak kode yang dieksekusi selama proses pengujian, semakin besar kemungkinan terdeteksinya bug atau kesalahan dalam program.

```
var a = 1;  
function x() {  
  var b = 2;  
  function y() {  
    var c = 3;  
    function z() {  
      var d = 4;  
    }  
    z();  
  }  
  y();  
}  
  
x();
```

☒ A ✓

☐ B

☐ C

☐ D

```

var a = 1;
function x() {
  var b = 2;
  function y() {
    var c = 3;
    function z() {
      var d = 4;
    }
    z();
  }
  y();
}
x();

```

☐ di mana saja dalam skrip!

☐ di mana saja di dalam fungsi `x()`

☒ di mana saja di dalam fungsi `y()`



☒ di mana saja di dalam fungsi `z()`



79. Imagery

Imagery dalam JavaScript merujuk pada kemampuan kita untuk membayangkan atau memvisualisasikan proses yang terjadi saat kode JavaScript dijalankan di browser atau lingkungan runtime JavaScript lainnya. Imagery membantu kita memahami bagaimana program berinteraksi dengan elemen HTML, mengambil data dari formulir, memanipulasi DOM, atau melakukan interaksi asinkron seperti mengambil data dari server atau menangani peristiwa.

Pertanyaan Kuis

Tanpa menempelkan ke konsol Anda, menurut Anda apa yang akan dicetak oleh kode ini?

```

var x = 1;

function addTwo() {
  x = x + 2;
}

addTwo();
x = x + 1;
console.log(x);

```

☐ 1

☐ 2

☐ 3

☒ 4



Pertanyaan Kuis

Tanpa menempelkan ke konsol Anda, menurut Anda apa yang akan dicetak oleh kode ini?

```
var x = 1;

function addTwo() {
  var x = x + 2;
}

addTwo();
x = x + 1;
console.log(x);
```

☐ 1

☒ 2



☐ 3

☐ 4

80. Variabel Global

Variabel global dalam JavaScript adalah variabel yang dideklarasikan di luar fungsi atau blok kode apapun sehingga dapat diakses dari mana saja dalam kode, baik itu di dalam atau di luar fungsi. Variabel global dikenal sebagai variabel yang bersifat global karena dapat diakses dari lingkup yang lebih luas dari pada variabel lokal.

81. Scope Recap

Apa yang telah Anda pelajari sejauh ini:

- Jika pengidentifikasi dideklarasikan dalam lingkup global, ini tersedia di mana saja.
- Jika pengidentifikasi dideklarasikan dalam function scope, itu tersedia dalam fungsi yang dideklarasikan (bahkan dalam fungsi yang dideklarasikan di dalam fungsi).
- Saat mencoba mengakses pengidentifikasi, JavaScript Engine pertama-tama akan melihat fungsi saat ini. Jika tidak menemukan apa pun, ia akan melanjutkan ke fungsi luar berikutnya untuk melihat apakah ia dapat menemukan pengidentifikasi di sana. Ini akan terus dilakukan hingga mencapai lingkup global.
- Pengidentifikasi global adalah ide yang buruk. Mereka dapat menyebabkan nama variabel yang buruk, nama variabel yang bertentangan, dan kode yang berantakan.

82. Lifting

Terkadang kode JavaScript Anda akan menghasilkan kesalahan yang mungkin tampak berlawanan dengan intuisi pada awalnya. Mengangkat adalah salah satu topik yang mungkin menjadi penyebab beberapa kesalahan rumit yang Anda debug.


```
sayHi("Julia");

function sayHi(name) {
  console.log(greeting + " " + name);
  var greeting;
}
```

- ☐ kesalahan
- ☐ belum diartikan
- ☐ Julia
- ☒ Julia yang tidak ditentukan
- ☐ batal
- ☐ null Julia
- ☐ NaN
- ☐ NaN Julia
- ☐ Halo Julia



Pertanyaan Kuis

Nilai apa yang akan dicetak ke konsol?

```
sayHi("Julia");

function sayHi(name) {
  console.log(greeting + " " + name);
  var greeting = "Hello";
}
```

- ☐ kesalahan
- ☐ belum diartikan
- ☐ Julia
- ☒ Julia yang tidak ditentukan
- ☐ batal
- ☐ null Julia
- ☐ NaN
- ☐ NaN Julia
- ☐ Halo Julia



```
function sayHi(name) {
  var greeting = "Hello";
  console.log(greeting + " " + name);
}

sayHi("Julia");
```

- ☐ kesalahan
- ☐ belum diartikan
- ☐ Julia
- ☐ Julia yang tidak ditentukan
- ☐ batal
- ☐ null Julia
- ☐ NaN
- ☐ NaN Julia
- ☒ Halo Julia



83. Lifting Recap

Apa yang telah Anda pelajari sejauh ini:

- JavaScript mengerek deklarasi fungsi dan deklarasi variabel ke bagian atas cakupan saat ini.
- Penugasan variabel tidak diangkat.
- Deklarasikan fungsi dan variabel di bagian atas skrip Anda, sehingga sintaks dan perilakunya konsisten satu sama lain.

84. Quiz: Build a Triangle (5-3)

```
14 // creates a line of * for a given length
15 function makeLine(length) {
16     var line = "";
17     for (var j = 1; j <= length; j++) {
18         line += "* ";
19     }
20     return line + "\n";
21 }
22
23 // your code goes here. Make sure you call makeLine() in your own code.
24 function buildTriangle(length) {
25     // Let's build a huge string equivalent to the triangle
26     var triangle = "";
27
28     //Let's start from the topmost line
29     var lineNumber = 1;
30
31     for(lineNumber=1; lineNumber<=length; lineNumber++){
32         // We will not print one line at a time.
33         // Rather, we will make a huge string that will comprise the whole triangle
34         triangle = triangle + makeLine(lineNumber);
35     }
36     return triangle;
37 }
38
39 // test your code by uncommenting the following line
40 // Note that the function buildTriangle() must return a string
41 // because the console.log() accepts a string argument
42 console.log(buildTriangle(10));
```

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

85. Function Expressions

Function expression adalah cara untuk mendefinisikan fungsi dalam bentuk ekspresi yang dievaluasi saat kode dieksekusi. Dalam function expression, fungsi dianggap sebagai nilai dan dapat disimpan dalam variabel atau properti objek. Berbeda dengan function declaration, di mana fungsi didefinisikan dengan kata kunci function di awal kode.

86. Patterns with Function Expressions

"Patterns with Function Expressions" merujuk pada penggunaan fungsi sebagai ekspresi dalam pola-pola pemrograman. Hal ini dapat digunakan dalam berbagai konteks, seperti:

- Higher-Order Functions: Dalam pemrograman fungsional, fungsi dapat diperlakukan seperti tipe data lainnya. Dalam hal ini, fungsi dapat digunakan

sebagai argumen untuk fungsi lain, atau dapat dikembalikan sebagai nilai dari fungsi itu sendiri.

- **Callback Functions:** Fungsi dapat digunakan sebagai callback function dalam JavaScript, yaitu fungsi yang dijalankan setelah suatu peristiwa terjadi. Contohnya, fungsi yang dijalankan ketika tombol di klik pada sebuah halaman web.
- **Anonymous Functions:** Fungsi juga dapat ditulis sebagai fungsi anonim atau tanpa nama, yang tidak perlu diberi nama dan dapat langsung digunakan sebagai argumen untuk fungsi lain atau ekspresi yang membutuhkannya.
- **Arrow Functions:** Fungsi panjang dapat ditulis secara lebih singkat menggunakan arrow function, yaitu fungsi yang ditulis dengan menggunakan tanda panah (=>) sebagai pengganti kata kunci function.

87. Function Expression Recap

"Function Expression Recap" adalah ringkasan atau ulasan kembali tentang penggunaan fungsi sebagai ekspresi dalam pemrograman.

- Function Expression adalah cara untuk menetapkan fungsi ke dalam variable.
- Fungsi yang ditetapkan ke dalam variabel dengan Function Expression dapat digunakan di mana saja seperti variabel biasa.
- Function Expression dapat ditulis sebagai fungsi anonim atau dengan nama. Fungsi anonim tidak memiliki nama dan dapat digunakan langsung sebagai argumen untuk fungsi lain atau ekspresi yang membutuhkannya.
- Function Expression dapat digunakan dalam Higher-Order Functions dan sebagai callback function.
- Arrow Function adalah cara lain untuk menulis Function Expression secara lebih singkat dan mudah dibaca.
- Dalam JavaScript, hoisting (mengangkat) tidak berlaku untuk Function Expression, sehingga variabel yang menampung Function Expression harus dideklarasikan terlebih dahulu sebelum digunakan.

88. Quiz: Laugh (5-4)

```
10  /*
11  *
12  */
13  // Declare an empty string
14  var sound = "";
15
16  // An anonymous function expression stored in the variable `laugh`
17  var laugh = function(num) {
18    // Iterate
19    for (var x = 0; x < num; x++) {
20      sound = sound + "ha";
21    }
22    sound = sound + "!";
23    return sound;
24  }
25
26  // It is essential that the function must return a string
27  console.log(laugh(3));
28
29
30
```

ha ha ha!

89. Quiz: Cry (5-5)

```
14
15 // your code goes here
16 var cry = function myFunction(){
17     var sound = "boohoo!";
18     return sound;
19 };
20
21 // Call the function using the variable name, not the function name
22 console.log(cry());
23
24 // This statement is also acceptable, but it won't print anything on your console
25 // cry();
```

boohoo!

90. Quiz: Inline (5-6)

```
13
14 // emotions() function definition
15 function emotions(myString, myFunc) {
16     console.log("I am " + myString + ", " + myFunc(2));
17 }
18
19 // Call the emotions() function with two arguments
20 // Argument 1 - "happy" string
21 // Argument 2 - an inline function expression
22 emotions ("happy", function(num) {
23     var sound = ""; // Local variable
24     //Iterate
25     for (var i = 0 ; i < num ; i++) {
26         sound = sound + "ha" ;
27     }
28     sound = sound + "!";
29     return sound;
30 })
```

I am happy, haha!

91. Lesson 5 Summary

Setelah kita belajar Fungsinya sekarang kita akan belajar array, lets'go.

92. Intro to Arrays

Array adalah salah satu struktur data yang penting dalam pemrograman. Array adalah kumpulan nilai atau objek yang disimpan dalam satu variabel dengan indeks yang teratur. Setiap nilai dalam array memiliki indeks numerik yang dimulai dari 0.

93. Membuat Array

Array berguna karena menyimpan beberapa nilai ke dalam struktur data tunggal yang terorganisir. Anda dapat mendefinisikan larik baru dengan mencantumkan nilai yang dipisahkan dengan koma di antara tanda kurung siku [].

Pertanyaan Kuis	
Pilih larik yang valid dari daftar di bawah ini.	
<input type="checkbox"/> ["pi" "pi" "pi" "pi"]	
<input checked="" type="checkbox"/> [33, 91, 13, 9, 23]	✓
<input checked="" type="checkbox"/> [nol, "", tidak terdefinisi, {}]	✓
<input checked="" type="checkbox"/> [3.14, "pi", 3, 1, 4, "Nyam, aku suka pai!"]	✓
<input type="checkbox"/> benar, 2, "Pai itu enak!"	
<input type="checkbox"/> [33; 91; 13; 9; 23]	
<input type="checkbox"/> {33, 91, 13, 9, 23}	

94. Array Index

Ingat bahwa elemen dalam array diindeks mulai dari posisi 0. Untuk mengakses elemen dalam array, gunakan nama array yang langsung diikuti dengan tanda kurung siku yang berisi indeks dari nilai yang ingin Anda akses.

Pertanyaan Kuis	
Lihatlah <code>donuts</code> larik berikut.	
<pre>var donuts = ["glazed", "chocolate frosted", "Boston cream", "powdered", "sprinkled", "maple", "coconut", "jelly"];</pre>	
Baris kode apa yang akan Anda gunakan untuk memilih <code>"coconut"</code> donat dari <code>donuts</code> larik?	
<input type="radio"/> donat[0]	
<input type="radio"/> donat[4]	
<input checked="" type="radio"/> donat[6]	✓
<input type="radio"/> donat[7]	

Pertanyaan Kuis	
Kami telah memutuskan untuk mengganti beberapa donat di dalam <code>donuts</code> array.	
<pre>var donuts = ["glazed", "chocolate frosted", "boston cream", "powdered", "sprinkled", "maple", "coconut", "jelly"];</pre> <pre>donuts[2] = "cinnamon twist"; donuts[4] = "salted caramel"; donuts[5] = "shortcake eclair";</pre>	
Seperti apa <code>donuts</code> tampilan array setelah perubahan berikut?	
<input type="radio"/> ["glazed", "chocolate frosted", "cinnamon twist", "salted caramel", "shortcake eclair", "maple", "coconut", "jelly"]	
<input type="radio"/> ["glazed", "cinnamon twist", "boston cream", "salted caramel", "shortcake eclair", "maple", "coconut", "jelly"]	
<input checked="" type="radio"/> ["glazed", "chocolate frosted", "cinnamon twist", "powdered", "salted caramel", "shortcake eclair", "coconut", "jelly"]	✓
<input type="radio"/> ["glazed", "cinnamon twist", "boston cream", "powdered", "salted caramel", "shortcake eclair", "coconut", "jelly"]	

95. Quiz: UdaciFamily (6-1)

```
13 // your code goes here
14 // You can create an array in different ways
15 // Method 1
16 // var udaciFamily = new Array("Julia", "James", "Yourname");
17
18 // Method 2
19 var udaciFamily = ["Julia", "James", "Yourname"];
20
21 // Iterate over the array elements
22 // We can find the size of an array using its `length` property
23 // Remember that the indexing of elements in an array starts from 0
24 for (var index = 0; index < udaciFamily.length; index++){
25     console.log(udaciFamily[index]);
26 }
27
```

Julia James Yourname

96. Quiz: Building the Crew (6-2)

```
13 var captain = "Mal";
14 var second = "Zoe";
15 var pilot = "Wash";
16 var companion = "Inara";
17 var mercenary = "Jayne";
18 var mechanic = "Kaylee";
19
20 // your code goes here
21 var crew = new Array(captain, second, pilot, companion, mercenary, mechanic);
22
23 // Another method
24 // var crew = [captain, second, pilot, companion, mercenary, mechanic];
25
26 // Print
27 console.log(crew);

```

['Mal', 'Zoe', 'Wash', 'Inara', 'Jayne', 'Kaylee']

97. Quiz: The Price is Right (6-3)

```
10 var prices = [1.23, 48.11, 90.11, 8.50, 9.99, 1.00, 1.10, 67.00];
11
12 // your code goes here
13 for (var index = 0; index < prices.length; index++){
14     //change the value of first element
15     if(index===0){
16         prices[index]=11;
17     }
18     //change the value of third element
19     else if(index===2){
20         prices[index]=33;
21     }
22     //change the value of seventh element
23     else if(index===6){
24         prices[index]=77;
25     }
26 }
27 console.log(prices);

```

[11, 48.11, 33, 8.5, 9.99, 1, 77, 67]

98. Array Properties and Methods

Array memiliki banyak properti dan metode yang dapat digunakan untuk memanipulasi dan mengelola data dalam array.

99. Length

Array.length adalah properti bawaan dari objek Array di JavaScript yang mengembalikan jumlah elemen dalam array. Properti ini dapat digunakan untuk mengakses dan mengetahui jumlah elemen dalam array, dan dapat digunakan dalam beberapa skenario pemrograman, seperti dalam iterasi melalui array menggunakan loop.

Quiz Question
What is the length of the following `inventory` array?

```
var inventory = [
  ["gold pieces", 25],
  ["belt", 4],
  ["ring", 1],
  ["shoes", 2]
];
```

☐ 1

☒ 4 ✓

☐ 8

☐ 32

100. Push

push() adalah metode bawaan dari objek Array di JavaScript yang digunakan untuk menambahkan satu atau lebih elemen baru ke akhir array dan mengembalikan panjang array yang baru. Metode ini memodifikasi array yang memanggil metodenya dengan menambahkan elemen baru ke akhirnya.

101. Pop

pop() adalah metode bawaan dari objek Array di JavaScript yang digunakan untuk menghapus elemen terakhir dari array dan mengembalikan elemen yang dihapus. Metode ini memodifikasi array yang memanggil metodenya dengan menghapus elemen terakhir dari array tersebut.

```
var donuts = ["glazed", "strawberry frosted", "powdered", "Boston creme"];
donuts.pop();
donuts.pop();
donuts.pop();
donuts.push("maple walnut");
donuts.pop();
donuts.push("sprinkled");
```

What does the `donuts` array look like after the following changes?

☐ ["sprinkled", "Boston Creme"]

☒ ["glazed", "sprinkled"] ✓

☐ ["maple walnut", "sprinkled"]

☐ ["sprinkled", "maple walnut"]

102. Splice

`splice()` adalah metode praktis lain yang memungkinkan Anda menambah dan menghapus elemen dari mana saja di dalam array.

Sementara `push()` dan `pop()` membatasi Anda untuk menambah dan menghapus elemen dari akhir array, `splice()` memungkinkan Anda menentukan lokasi indeks untuk menambahkan elemen baru, serta jumlah elemen yang ingin Anda hapus (jika ada).

Pertanyaan Kuis

Kami telah memutuskan untuk mengganti beberapa donat di dalam `donuts` array.

```
var donuts = ["cookies", "cinnamon sugar", "creme de leche"];  
  
donuts.splice(-2, 0, "chocolate frosted", "glazed");
```

Seperti apa `donuts` tampilan array setelah perubahan berikut?

- ☒ ["cookies", "cokelat beku", "glazed", "gula kayu manis", "creme de leche"] ✓
- ☐ ["coklat beku", "mengkilap", "kue", "gula kayu manis", "creme de leche"]
- ☐ ["coklat beku", "mengkilap"]
- ☐ ["cookies", "cinnamon sugar", "creme de leche", "cokelat frosted", "glazed"]

103. Quiz: Rainbow Colors (6-4)

```
3  *  
4  * Use only the splice() method to modify the rainbow variable:  
5  * - remove "Blackberry"  
6  * - add "Yellow" and "Green"  
7  * - add "Purple"  
8  */  
9  
10 var rainbow = ['Red', 'Orange', 'Blackberry', 'Blue'];  
11  
12 // your code goes here  
13 // From index position 2, remove 1 element. Afterwards, add all the elements mentioned in t  
14 rainbow.splice(2, 1, "Yellow", "Green");  
15  
16 // From index position 5, remove 0 elements. This means, no removal of any element.  
17 // Afterwards, add all the elements mentioned in the argument.  
18 rainbow.splice(5, 0, "Purple");  
19  
20 // Did you noticed that the above method call is equivalent to adding an element at a speci  
21  
22 //Print  
23 console.log(rainbow);
```

```
['Merah', 'Oranye', 'Kuning', 'Hijau', 'Biru', 'Ungu']
```


104. Kuis: Piala Quidditch (6-5)

```
11
12 // your code goes here
13
14 var team = ["Oliver Wood", "Angelina Johnson", "Katie Bell", "Alicia Spinnet", "George Weasley", "Fred Weasley", "George Weasley"];
15 console.log(hasEnoughPlayers(team));
16 var team = ["Oliver Wood", "Angelina Johnson", "Katie Bell", "Alicia Spinnet", "George Weasley", "Fred Weasley", "George Weasley"];
17
18 function hasEnoughPlayers(arrayInstance) {
19     if (arrayInstance.length >= 7)
20         return true; // true is a boolean value. Do not return a string "true"
21     else
22         return false; // false is also a boolean value. Do not return a string "false"
23 }
24 console.log(hasEnoughPlayers(team));
```

betul betul

105. Quiz: Join the Crew (6-6)

```
7
8 var captain = "Mal";
9 var second = "Zoe";
10 var pilot = "Wash";
11 var companion = "Inara";
12 var mercenary = "Jayne";
13 var mechanic = "Kaylee";
14
15 var crew = [captain, second, pilot, companion, mercenary, mechanic];
16
22 var doctor = "Simon";
23 var sister = "River";
24 var shepherd = "Book";
25
26 // your code goes here
27 // your code goes here
28 crew.push(doctor); // "Simon" gets added at the end of the array
29 // We have intentionally not captured the return of the push() method
30
31 crew.push(sister); // "River" gets added at the end of the array
32 crew.push(shepherd); // "Book" gets added at the end of the array
33
34 console.log(crew);
```

['Mal', 'Zoe', 'Cuci', 'Inara', 'Jayne', 'Kaylee', 'Simon', 'Sungai', 'Buku']

106. Quiz: Checking Documents (6-7)

Pertanyaan Kuis

Gunakan [Dokumentasi MDN](#) untuk menentukan metode mana yang terbaik untuk membalikkan elemen dalam larik ini:

```
var reverseMe = ["h", "e", "l", "l", "o"];
```

☐ pop()

☐ dorongan()

☒ balik()



☐ menggeser()

☐ sambatan()

Pertanyaan Kuis

Apa metode array terbaik untuk mengurutkan elemen dalam array ini:

```
var sortMe = [2, 1, 10, 7, 6];
```

☐ concat()

☐ bergabung()

☐ unshift()

☒ menyortir()



☐ mengatur()

Pertanyaan Kuis

Pertimbangkan larik berikut, `removeFirstElement`:

```
var removeFirstElement = ["a", "b", "c"];
```

Katakanlah Anda ingin memodifikasi (yaitu, *mutate*) `removeFirstElement` dengan menghapus elemen *pertama* di dalamnya. Metode mana yang dapat Anda gunakan?

☐ pop()

☐ dorongan()

☒ menggeser()



☐ unshift()

☒ sambatan()



☐ mengiris()

Pertanyaan Kuis

Metode apa yang terbaik untuk mengubah array ini menjadi string?

```
var turnMeIntoAString = ["U", "d", "a", "c", "i", "t", "y"];
```

☒ bergabung()



☐ concat()

☐ mengisi()

☐ copyWithin()

107. Loop Array

Loop Array adalah sebuah teknik dalam pemrograman komputer yang digunakan untuk mengakses setiap elemen dalam sebuah array secara berulang-ulang. Dalam loop array, perintah atau kode akan dieksekusi untuk setiap elemen dalam array, sehingga memungkinkan pengguna untuk melakukan operasi atau manipulasi data pada setiap elemen secara efisien.

```
my_array = [1, 2, 3, 4, 5]

# Loop array menggunakan for loop
for element in my_array:
    print(element)
```

108. forEach Loop

Fungsi yang Anda berikan ke `forEach()` metode dapat menggunakan hingga tiga parameter. Di dalam video, ini disebut `element`, `index`, dan `array`, tetapi Anda dapat menamainya sesuai Anda.

Metode ini `forEach()` akan memanggil fungsi ini satu kali untuk setiap elemen dalam array (maka nama `forEach`.) Setiap kali, ia akan memanggil fungsi dengan argumen yang berbeda. Parameter `element` akan mendapatkan nilai dari elemen array. Parameter `index` akan mendapatkan indeks elemen (dimulai dengan nol). Parameter `array` akan mendapatkan referensi ke seluruh larik, yang berguna jika Anda ingin memodifikasi elemen.

```
words = ["cat", "in", "hat"];
words.forEach(function(word, num, all) {
  console.log("Word " + num + " in " + all.toString() + " is " + word);
});
```

Cetakan:

Kata 0 pada kucing,dalam,topi adalah kucing

Kata 1 pada kucing,dalam,topi dalam

Kata 2 dalam kucing,dalam,topi adalah topi

109. Quiz: Other Types of Rounds (6-8)

```
9 var test = [12, 929, 11, 3, 199, 1000, 7, 1, 24, 37, 4,
10 19, 300, 3775, 299, 36, 209, 148, 169, 299,
11 6, 109, 20, 58, 139, 59, 3, 1, 139
12 ];
13
14 // Write your code here
15 // You can test your code with a smaller array first
16 // var test = [12, 29, 11, 3];
17
18 // Write your code here
19 test.forEach(function(item, index){
24 if(item%3===0){
25   // The commented statements below would not be able to change the value of item
26   // because the `item` is a COPY of actual element, while actual is test[index]
27   // item = item + 100;
28   // console.log("The updated element is "+item+" and actual is "+test[index]);
29   test[index]=test[index]+100;
30 }
31 //else
32 // console.log("Element "+item+" is not divisible by 3");
33 });
34 console.log(test);
```

```
[ 112, 929, 11, 103, 199, 1000, 7, 1, 124, 37, 4, 19, 400, 3775, 299, 136, 209,
148, 169, 299, 106, 109, 20, 58, 139, 59, 103, 1, 139 ]
```

110. Map

map adalah sebuah metode bawaan dari tipe data Array pada bahasa pemrograman JavaScript. Metode ini digunakan untuk mengubah setiap elemen dalam array dengan mengaplikasikan sebuah fungsi pada setiap elemen, dan kemudian mengembalikan array baru dengan elemen-elemen yang telah diubah.

Contoh sederhana dari penggunaan map pada JavaScript adalah sebagai berikut:

```
const numbers = [1, 2, 3, 4, 5];

const doubleNumbers = numbers.map(function(number) {
  return number * 2;
});

console.log(doubleNumbers);
```

111. Quiz: I Got the Bill (6-9)

```
4
5 var bills = [50.23, 19.12, 34.01,
6             100.11, 12.15, 9.90, 29.11, 12.99,
7             10.00, 99.22, 102.20, 100.10, 6.77, 2.22
8 ];
9 var bills = [50.23, 19.12, 34.01,
10            100.11, 12.15, 9.90, 29.11, 12.99,
11            10.00, 99.22, 102.20, 100.10, 6.77, 2.22
12 ];
13 var totals = bills.map(function (element) {
14     element *= 1.15;
15     element = element.toFixed(2);
16     element = Number(element);
17     return element;
18 });
19
20
21
22 /*
23 // Method 2
24 var totals = bills.map(function(element) {
25     element *= 1.15;
26     return Number(element.toFixed(2));
27 });
28 */
29 console.log(totals);
30
31
```

```
[ 57.76, 21.99, 39.11, 115.13, 13.97, 11.38, 33.48, 14.94, 11.5, 114.1, 117.53,
115.11, 7.79, 2.55 ]
```

112. Array in Array

Array in Array atau Array dalam Array, juga dikenal sebagai nested array, adalah struktur data di mana sebuah array dijadikan sebagai elemen dalam array lainnya. Dalam nested array, setiap elemen dalam array induk adalah array yang berdiri sendiri dengan elemen-elemennya sendiri.

```
const nestedArray = [[1, 2], [3, 4, 5], [6, 7, 8, 9]];

console.log(nestedArray[0]); // [1, 2]
console.log(nestedArray[1][1]); // 4
console.log(nestedArray[2][3]); // 9
```

113. 2D Donut Arrangement

2D Donut Arrangement adalah susunan matriks dua dimensi yang memiliki bentuk seperti donat. Susunan ini terdiri dari angka-angka dari 1 sampai n^2 yang dibagi-bagi secara merata dan diposisikan dalam bentuk lingkaran, sehingga menghasilkan bentuk matriks yang berbentuk seperti donat.

Contoh dari 2D Donut Arrangement adalah sebagai berikut:

```
7 8 9 10 11
6 19 20 21 12
5 18 23 22 13
4 17 16 15 14
3 2 1 24 25
```

114. Quiz: Nested Numbers (6-10)

```
10
11- var numbers = [
12     [243, 12, 23, 12, 45, 45, 78, 66, 223, 3],
13     [34, 2, 1, 553, 23, 4, 66, 23, 4, 55],
14     [67, 56, 45, 553, 44, 55, 5, 428, 452, 3],
15     [12, 31, 55, 445, 79, 44, 674, 224, 4, 21],
16     [4, 2, 3, 52, 13, 51, 44, 1, 67, 5],
17     [5, 65, 4, 5, 5, 6, 5, 43, 23, 4424],
18     [74, 532, 6, 7, 35, 17, 89, 43, 43, 66],
19     [53, 6, 89, 10, 23, 52, 111, 44, 109, 80],
20     [67, 6, 53, 537, 2, 168, 16, 2, 1, 8],
21     [76, 7, 9, 6, 3, 73, 77, 100, 56, 100]
22 ];
23
24 // your code goes here
25 // Let i represents a sub-array
26- for(var i=0; i<numbers.length; i++){
27
28     // Let j represents the length of a sub-array
29-     for(var j=0; j<numbers[i].length; j++){
30         // numbers[i][j]
31     }
32 }
```

```
[ [ 'ganjil', 'genap', 'ganjil', 'genap', 'ganjil', 'ganjil', 'genap', 'genap', 'ganjil', 'ganjil'], [ 'genap',
'genap ', 'ganjil', 'ganjil', 'ganjil', 'genap', 'genap', 'ganjil', 'ganjil'], [ 'ganjil', 'genap',
'ganjil', 'ganjil', ' ', 'genap', 'ganjil', 'ganjil', 'genap', 'genap', 'ganjil'], [ 'genap', 'ganjil', 'ganjil',
'ganjil', 'ganjil', 'genap ', 'genap', 'genap', 'genap', 'ganjil'], [ 'genap', 'genap', 'ganjil', 'genap',
'ganjil', 'ganjil', 'genap', 'ganjil', 'ganjil', 'ganjil' ], [ 'ganjil', 'ganjil', 'genap', 'ganjil', '
genap', 'ganjil', 'ganjil', 'ganjil', 'genap' ], [ 'genap', 'genap', 'ganjil', 'ganjil', 'ganjil',
'ganjil', ' ganjil', 'ganjil', 'genap' ], [ 'ganjil', 'genap', 'ganjil', 'genap', 'ganjil', 'genap', 'ganjil',
'genap', 'ganjil', ' genap'], [ 'ganjil', 'genap', 'ganjil', 'ganjil', 'genap', 'genap', 'genap', 'ganjil',
'genap' ], [ 'genap', 'ganjil', 'ganjil', 'genap', 'ganjil', 'ganjil', 'ganjil', 'genap', 'genap', 'bahkan' ] ]
```

115. Objects in Code

Dalam pemrograman, object adalah sebuah tipe data kompleks yang terdiri dari kumpulan pasangan nama dan nilai, yang disebut sebagai properti (atau dalam bahasa pemrograman lainnya disebut sebagai fields atau attributes). Properti-properti ini dapat berupa nilai primitif seperti string, number, dan boolean, atau bahkan dapat berupa objek lain, termasuk fungsi (yang dalam hal ini sering disebut sebagai method).

Dalam JavaScript, objek dapat didefinisikan dengan menggunakan sintaksis objek literal. Contoh sederhana dari objek literal adalah sebagai berikut:

```
const person = {
  name: "John Doe",
  age: 30,
  address: {
    street: "123 Main St",
    city: "Anytown",
    state: "CA",
    zip: "12345"
  },
  sayHello: function() {
    console.log("Hello!");
  }
};

console.log(person.name); // "John Doe"
console.log(person.address.city); // "Anytown"
person.sayHello(); // "Hello!"
```

116. Quiz: Umbrella (7-1)

```
5  */
6
7- var umbrella = {
8    color: "pink",
9    isOpen: true,
10-   open: function() {
11-       if (umbrella.isOpen === true) {
12-           return "The umbrella is already opened!";
13-       } else {
14-           umbrella.isOpen = true;
21-           return "Julia opens the umbrella!";
22-       }
23-   },
24-   // your code goes here
25-   // method definition within the object definition
26-   close: function() {
27-       if (umbrella.isOpen === true) {
28-           umbrella.isOpen = false;
29-           return "Julia closes the umbrella!";
30-       } else {
31-           return "The umbrella is already closed!";
32-       }
33-   }
34- };
```

117. Objek

Objek pada JavaScript adalah kumpulan dari properti (atau key) dan nilai (atau value) yang terkait satu sama lain. Properti adalah nama atau identifier yang diberikan pada nilai dalam objek. Objek pada JavaScript dapat merepresentasikan benda nyata seperti orang, mobil, atau rumah, atau konsep abstrak seperti waktu, tanggal, dan warna.

```
var sister = {
    name: "Sarah",
    age: 23,
    parents: ["Alice", "Andy"],
    siblings: ["Julia"],
    favoriteColor: "purple",
    pets: true
};
```

118. Object Literals

Object Literals adalah cara untuk membuat objek pada JavaScript dengan menggunakan sintaksis objek literal {}. Sintaksis ini memungkinkan kita untuk membuat objek secara langsung tanpa harus menggunakan constructor function.

```
var sister = {  
  name: "Sarah",  
  age: 23,  
  parents: [ "alice", "andy" ],  
  siblings: ["julia"],  
  favoriteColor: "purple",  
  pets: true,  
  paintPicture: function() { return "Sarah paints!"; }  
};  
  
sister.paintPicture();
```

Pengembalian : "Sarah melukis!"

119. Naming Convention

Pertanyaan Kuis

Diberi objek berikut `user`, apa yang akan Anda gunakan untuk mencetak nilai properti `email`?

```
var user = {  
  email: "user@example.com",  
  firstName: "first",  
  lastName: "last"  
};
```

☐ `console.log(pengguna[email]);`

☐ `console.log(email);`

☒ `console.log(pengguna.email);`



☒ `console.log(pengguna["email"]);`



Pertanyaan Kuis

Pilih potongan kode yang membuat objek yang mendeskripsikan Honda Civic merah:



`var mobil = { pabrikan: "honda", model: "civic", class: "compact", color: "red" };`



`var pabrikan = "honda"; var model = "sipil"; var kelas = "kompak"; var warna = "merah";`



`var mobil = { color: "red" , pabrikan: "honda", model: "civic", class: "compact" };`



`var mobil = ["merah", "honda", "sipil", "kompak"];`

120. Object Summary

Objek adalah salah satu struktur data terpenting dalam JavaScript. Bersiaplah untuk melihat mereka di mana-mana!

Mereka memiliki properti (informasi tentang objek) dan metode (fungsi atau kemampuan yang dimiliki objek). Objek adalah tipe data yang sangat kuat dan Anda akan melihatnya di semua tempat saat bekerja dengan JavaScript, atau bahasa pemrograman berorientasi objek lainnya.

```
var myObj = {  
  color: "orange",  
  shape: "sphere",  
  type: "food",  
  eat: function() { return "yummy" }  
};  
  
myObj.eat(); // method  
myObj.color; // property
```

121. Quiz: Menu Items (7-2)

```
8 // your code goes here
9 // your code goes here
10 var breakfast = {
11     name: "The Lumberjack",
12     price: "$9.95",
13     ingredients: ["eggs", "sausage", "toast", "hashbrowns", "pancake"],
14 };
15 console.log(breakfast);
```

```
{ nama: 'The Lumberjack', harga: '$9.95', bahan-bahan: [
  'telur', 'sisis', 'roti bakar', 'hashbrowns', 'pancake'
] }
```

122. Quiz: Bank Account 1 (7-3)

```
8 var savingsAccount = {
9     balance: 1000,
10    interestRatePercent: 1,
11    deposit: function addMoney(amount) {
12        if (amount > 0) {
13            savingsAccount.balance += amount;
14        }
15    },
16    withdraw: function removeMoney(amount) {
17        var verifyBalance = savingsAccount.balance - amount;
18        if (amount > 0 && verifyBalance >= 0) {
19            savingsAccount.balance -= amount;
20        }
21    },
22    // your code goes here
23    printAccountSummary: function() {
24        return "Welcome!\nYour balance is currently $" + savingsAccount.balance;
25    }
26 };
27
28 console.log(savingsAccount.printAccountSummary());
29
```

Selamat datang! Saldo Anda saat ini \$1000 dan suku bunga Anda adalah 1%.

123. Quiz: Bank Account 2 (7-4)

Pertanyaan Kuis	
Manakah dari berikut ini cara yang valid untuk mengakses properti dan memanggil metode dari <code>savingsAccount</code> objek?	
<input checked="" type="checkbox"/>	<code>tabunganAccount.saldo;</code> ✓
<input type="checkbox"/>	<code>rekening tabungan[saldo];</code>
<input checked="" type="checkbox"/>	<code>rekening tabungan["saldo"];</code> ✓
<input type="checkbox"/>	<code>savingAccount."saldo";</code>
<input type="checkbox"/>	<code>savingAccount.withdraw = 50;</code>
<input type="checkbox"/>	<code>savingAccount.removeMoney = 50;</code>
<input checked="" type="checkbox"/>	<code>savingAccount.withdraw(50);</code> ✓
<input type="checkbox"/>	<code>savingAccount.removeMoney(50);</code>

124. Quiz: Facebook Friends (7-5)

```

6
7 // your code goes here
8 var facebookProfile = {
9   name: "Udacious",
10  friends: 25,
11  messages: ["Message 1", "Message 2", "Message 3", "Message 4"],
12  postMessage: function(message){
13    facebookProfile.messages.push(message);
14  },
15  deleteMessage: function(index){
16    // In the following splice() method call,
17    // - argument 1 = index from where the element has to be deleted
18    // - argument 2 = count of elements to be deleted
19    facebookProfile.messages.splice(index, 1);
20  },
21  addFriend: function(){
22    facebookProfile.friends = facebookProfile.friends + 1;
23    //friends += 1; // this statement is equivalent to the above
24  },
25  removeFriend: function(){
26    if(facebookProfile.friends > 0)
27      facebookProfile.friends = facebookProfile.friends - 1;
28  }
29 };

```

125. Quiz: Donut Review (7-6)

```
12
13 // This is an array of objects.
14 var donuts = [
15   { type: "Jelly", cost: 1.22 },
16   { type: "Chocolate", cost: 2.45 },
17   { type: "Cider", cost: 1.59 },
18   { type: "Boston Cream", cost: 5.99 }
19 ];
20
21 // your code goes here
22 donuts.forEach(function(donut){
23
24   // donut represents a single element of donuts array
25   // donut is an object, therefore you can access its properties using a dot notation
26   console.log(donut.type+" donuts cost $" + donut.cost+" each");
27 });
```

Donat jeli seharga \$1,22 per buah Donat coklat seharga \$2,45 per buah Cider donat seharga \$1,59 per buah Boston Cream donat seharga \$5,99 per buah

Pengantar JavaScript

Lanjutkan di mana Anda tinggalkan

Objek

Melanjutkan →

Pelajaran

- **Apa itu JavaScript?**
Pelajari sejarah JavaScript dan segera mulai menulis kode Anda menggunakan konsol JavaScript. ✓
- **Tipe Data & Variabel**
Belajar merepresentasikan data dunia nyata menggunakan variabel JavaScript, dan membedakan berbagai tipe data dalam bahasa tersebut. ✓
- **Persyaratan**
Pelajari cara menambahkan logika ke program JavaScript Anda menggunakan pernyataan bersyarat. ✓
- **Loop**
Manfaatkan kekuatan pengulangan JavaScript untuk mengurangi duplikasi kode dan mengotomatiskan tugas berulang. ✓
- **Fungsi**
Selami dunia fungsi JavaScript. Belajar memanfaatkan kekuatan mereka untuk merampingkan dan mengatur program Anda. ✓
- **Array**
Pelajari cara menggunakan Array untuk menyimpan data kompleks dalam program JavaScript Anda. ✓
- **Objek**
Temui struktur data JavaScript berikutnya: Objek. Pelajari cara menggunakannya untuk menyimpan data kompleks bersama Array. ✓

Bukti bahwasannya telah menyelesaikan tugas rangkuman materi JavaScript Fundamentals