

Dasar Pemrograman Web

Pertemuan Ke-8 (JavaScript)

Noor Ifada

Email: noor.ifada@trunojoyo.ac.id

Scopus: [56590032100](#)

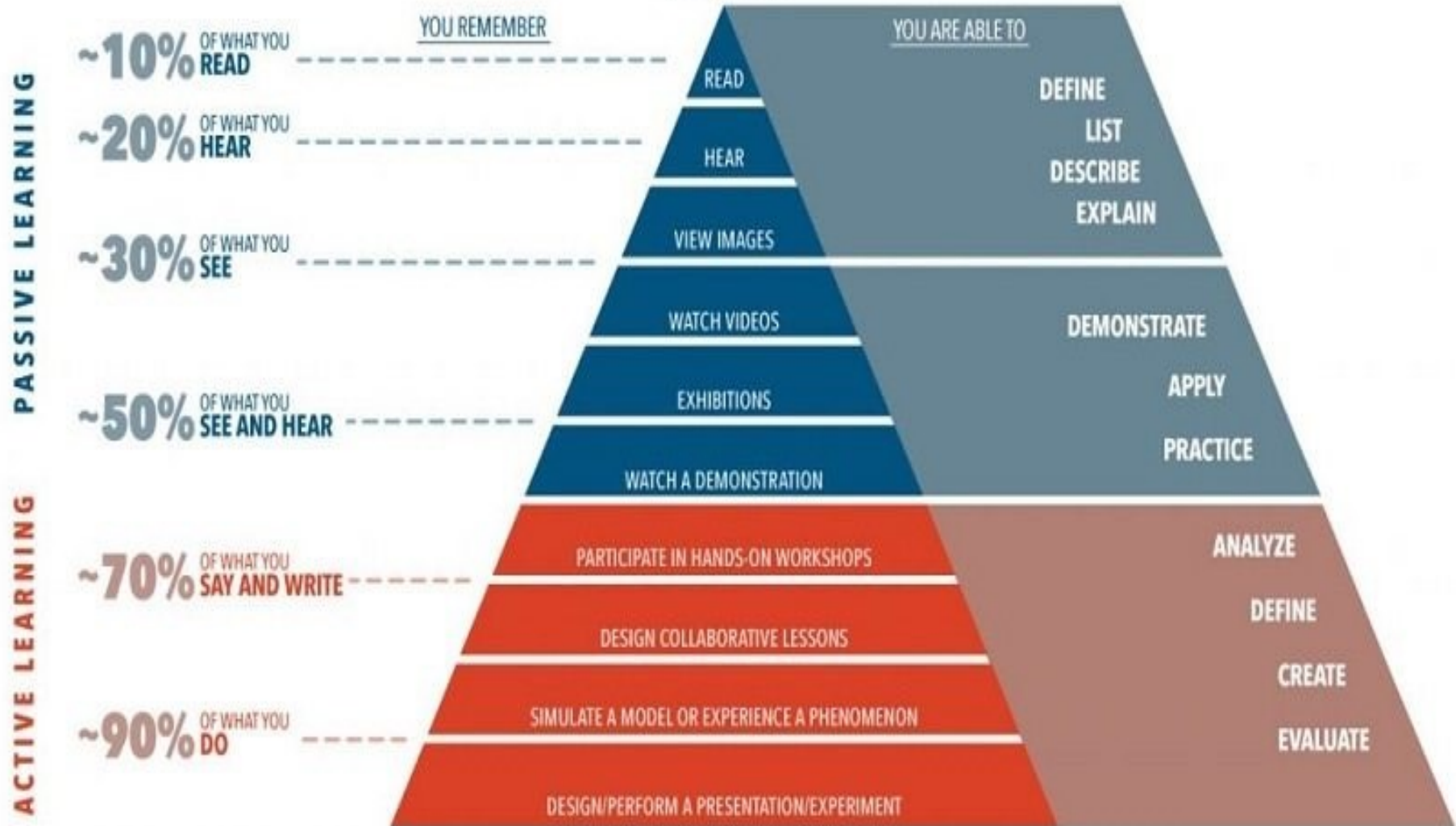
Google Scholar: [Noor Ifada](#)

ResearchGate: [Noor-Ifada](#)

Repository: [Trunojoyoan](#)

CONE OF EXPERIENCE

EDGAR DALE



<https://elearningindustry.com/>

SOFT SKILLS FOR STUDENTS

6 SKILLS TO HELP YOU SUCCEED



START

1 Problem Solving

Do you see problems as **interesting opportunities**? Problem solvers anticipate potential stumbling blocks and act to prevent them or mitigate their effects.



TIP: You don't always need to start by trying to solve the problem. First, aim to understand the root of the problem.



2 Creativity

Are you able to **think outside of the box** even when you're not asked to? Everyone has creative abilities. Creative thinking skills help you to look at problems and situations from a fresh perspective.

3 Communication

Are you an **effective communicator and active listener**? Communication skills will help you understand others and be better understood in return. This can help make each interaction you have in both your work and social life a more positive experience.



4 Time Management

Are you an **organized person who meets deadlines**? Good time management skills enable you to do more in less time, helping you reduce stress and maintain a better work-life balance.

TIP: Prioritize your tasks and delegate a time limit to how long you spend on each one.

5 Stress Management

Can you **identify and tackle your stress triggers**? Being able to manage stress allows you to break the hold it can have on your life. This will make you happier, healthier, and more productive in return.

TIP: Set limits and learn to say no to requests that would create excessive stress in your life.



6 Teamwork

Are you a **team player who collaborates well with others**? Teamwork requires leadership and goal orientation. Whether in school or a job, the better you work within a team, the better your work will be!

TIP: Always clarify roles, responsibilities and accountability with your team members.




END



educations.com

Tabu Bagi Mahasiswa Teknik Informatika

- Tidak dapat beradaptasi dengan (perkembangan) teknologi
- Tidak dapat membaca atau memahami instruksi
- Tidak dapat membuat program



**LEARNING IS NEVER
DONE WITHOUT
ERRORS AND DEFEAT**

VLADIMIR LENIN

PICTUREQUOTES.COM



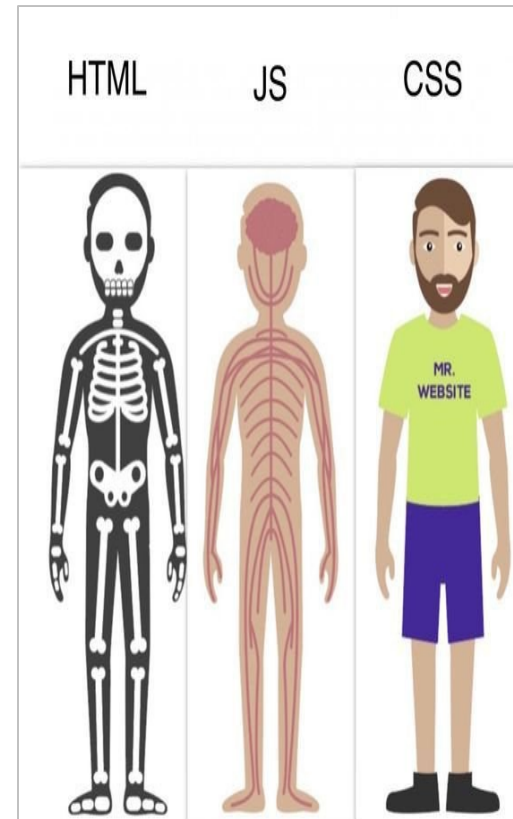
PICTUREQUOTES

Sub Pokok Bahasan

- *JavaScript*
- Contoh *JavaScript* Sederhana
- Lokasi penempatan *JavaScript*
- Console: *Chrome DeveloperTools*
- Variabel
- Operator
- Struktur Kondisi
- Struktur Perulangan
- Fungsi (*Function*)
- *Built-in Objects*
- *RegExp*

JavaScript

- Merupakan Bahasa Pemrograman *client-side* yang digunakan untuk mengatur perilaku halaman *web* pada *browser*
 - Adalah *object-based language*
 - Memiliki *objects*, *methods* dan *properties*
 - Namun tidak seperti Java:
 - *Object* pada JavaScript bukan merupakan *class*
 - Variabel tidak perlu dideklarasikan tipe datanya
- Merupakan *Dynamic language (weakly typed)*
 - Tidak perlu mendeklarasi tipe data variabel & parameter
 - Sama seperti Python
 - Berbeda dengan Java/C# (*strongly typed languages*)
- Bersesuaian dengan *C type syntax*
 - Mirip dengan Java dan C#, sedikit berbeda dari Python



Contoh JavaScript Sederhana

```
<p>  
  <script> // tag untuk memulai JavaScript  
    document.write("Membuat JavaScript Sederhana!");  
    /* tambahkan kode JavaScript  
       yang lain di sini  
       */  
  </script>  
</p>
```

Membuat JavaScript Sederhana!

Challenge #1

Apa (nama) *tag* yang digunakan untuk membuat skrip JavaScript?

Implementasikan skrip JavaScript yang ditampilkan pada *Slide #6* (“Contoh JavaScript Sederhana”) ke dalam sebuah dokumen HTML!

Demonstrasikan hasil yang ditampilkan dokumen HTML yang berisikan skrip JavaScript pada *web browser*!

Lokasi JavaScript: Elemen <BODY>

- *JavaScript* akan dieksekusi setiap kali halaman *web* dibuka

```
<html>
  <head>
  </head>
  <body>
    <script>
      . . .
    </script>
  </body>
</html>
```

Lokasi JavaScript: Elemen <HEAD>

- JavaScript akan dipanggil ketika ada *event* yang berfungsi sebagai *trigger*

```
<html>
  <head>
    <script>
      . . .
    </script>
  </head>
  <body>
  </body>
</html>
```

Lokasi JavaScript: File Eksternal

File HTML

```
<html>
  <head>
  <head>
  <body>
    <p>
      <script src="eksternal.js">
      </script>
    </p>
  </body>
</html>
```

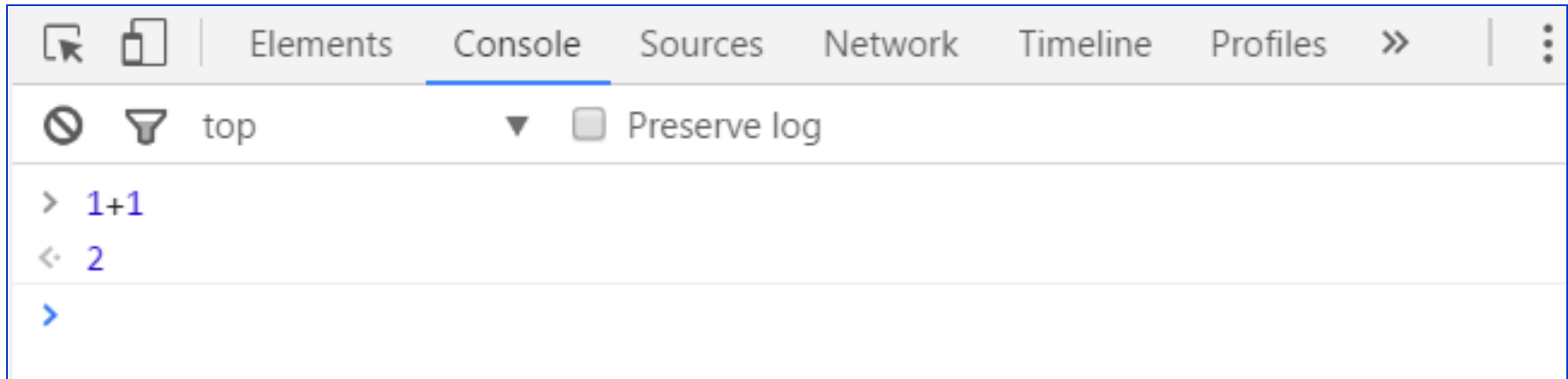
File JavaScript (nama file: **eksternal.js**)

```
document.write("Teks ini ditampilkan dengan menggunakan file  
eksternal JavaScript");
```

Teks ini ditampilkan dengan menggunakan file eksternal JavaScript

Console: Chrome DeveloperTools

- Command+Option+i on OS X
- F12 or Ctrl+Shift+I on Windows



NOTE: Gunakan tab “Console” dalam Developer Tools untuk mempermudah mengetes skrip JavaScript

- Tombol **SHIFT+ENTER** berfungsi untuk berpindah baris
- Tombol **ENTER** berfungsi untuk mengeksekusi skrip JavaScript

Challenge #2

Dengan menggunakan acuan *Slide #11*:

Demonstrasikan cara menggunakan *DeveloperTools* untuk mengetes skrip JavaScript yang ditampilkan pada *Slide #6*!

Note: Penggunaan *DeveloperTools* juga akan diimplementasikan dalam **Challenge #3** hingga **#10**

Variabel

- Tipe data: *Numeric, String, Array, Boolean, Null, Undefined*
- Aturan penamaan:
 - adalah *case-sensitive*
 - harus dimulai dengan suatu **huruf**, karakter **garis bawah** (“_”), ataupun simbol dollar (“\$”)
 - tidak menggunakan kata-kata khusus dalam JavaScript seperti **abstract, break, case**, dan lain-lain (daftar lengkap dapat dilihat di https://www.w3schools.com/js/js_reserved.asp)

Type Data	Contoh
Numerik	var angka1 = 17; angka2 = 3.78;
String	var nama = 'eve';
Array	var arrDayNames = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]; arrDayNames[2] = "Tuesday";
Boolean	var status = true;
Null	var huruf = null;
Undefined	var huruf1; huruf2 = undefined;

Operator

Aritmatika (*Arithmetic*)

Simbol	Deskripsi	Contoh (x = 10)	Hasil
+	Penjumlahan	x + 5	15
-	Pengurangan	x - 5	5
*	Perkalian	x * 5	50
/	Pembagian	x / 5	2
%	Modulus	x % 5	0
++	Increment (menambahkan nilai variabel dengan 1)	x++	11
--	Decrement (mengurangi nilai variabel dengan 1)	x--	9

Penugasan (*Assignment*)

Simbol	Contoh	Ekivalensi
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Logika (*Logical*)

Operator	Nama	Deskripsi	Contoh (x = 1 dan y = 2)
&&	Logika AND	Bernilai true jika kedua kondisi (di sisi kiri dan kanan operator) adalah true	(x < 2 && y > 1) bernilai true
??	Logika OR	Bernilai true jika salah satu kondisi (di sisi kiri atau kanan operator) adalah true	(x < 2 && y > 2) bernilai true
!	Logika NOT	Kebalikan dari nilai kondisi	!(x > y) bernilai true

Operator [2]

Perbandingan		
Operator	Deskripsi	Contoh
==	Sama dengan	1==2 bernilai false 2==2 bernilai true 2=="2" bernilai true
===	Sama nilai dan sama tipe data	2===2 bernilai true 2=== "2" bernilai false
!=	Tidak sama dengan	1!=2 bernilai true 2!=2 bernilai false
!==	Tidak sama nilai atau tidak sama tipe data	1!==2 bernilai false 2!==2 bernilai false 2!== "2" bernilai true
>	Lebih besar dari	1>2 bernilai false 2>2 bernilai false 3>2 bernilai true
<	Lebih kecil dari	1 < 2 bernilai true 2 < 2 bernilai false 3 < 2 bernilai false
>=	Lebih besar atau sama dengan	1 >= 2 bernilai false 2 >= 2 bernilai true 3 >= 2 bernilai true
<=	Lebih kecil atau sama dengan	1 <= 2 bernilai true 2 <= 2 bernilai true 3 <= 2 bernilai false
?	Operator kondisional atau ternary (memberikan nilai pada suatu variabel berdasarkan suatu kondisi)	var keterangan = (angka < 55) ? "Tidak Lulus":"Lulus"; angka = 50; Keterangan bernilai "Tidak Lulus"

Operator [3]

<i>Bitwise</i>					
Operator	Deskripsi	Contoh	Ekivalensi biner (Contoh: 4 bit, Riil: 32 bit)	Hasil	Desimal
&	AND	2 & 1	0010 & 0001	0000	0
	OR	2 1	0010 0001	0011	3
~	NOT	~ 2	~0010	1101	13
^	XOR	2 ^ 1	0010 ^ 0001	0011	3
<<	Geser kiri dan isi bit paling kanan dengan sejumlah bit 0	2 << 1	0010 << 1	0100	4
>>	Geser kanan dan isi bit paling kiri dengan sejumlah bit yang sama dengan sebelumnya	-2 >> 1	1110 >> 1	1111	-1
>>>	Geser kanan dan isi bit paling kiri dengan sejumlah bit 0	2 >>> 1	0010 >>> 1	0001	1

String		
Operator	Deskripsi	Contoh
+	Konkatenasi atau menggabungkan string	firstName = "Almira"; lastName = "Wijaya"; name = firstName + " " + lastName; name bernilai "Almira Wijaya"
+=	Konkatenasi atau menggabungkan string	txt1="Hello "; txt1 += "World" txt1 bernilai "Hello World"

Struktur Kondisi: IF

- Struktur **IF** digunakan untuk memastikan bahwa suatu perintah akan dieksekusi ketika kondisi bernilai *true*. Tambahan struktur **ELSE** diperlukan jika suatu eksekusi akan dilakukan ketika kondisi bernilai *false*. Sedangkan struktur **ELSE IF** digunakan untuk menambahkan kondisi baru ketika kondisi bernilai *false*

```
var warna = "red";  
if(warna == "green")  
{  
    warna = "hijau";  
}  
else if (warna == "blue")  
{  
    warna = "biru";  
}  
else  
{  
    warna = "merah";  
}  
warna;
```

- Coba cek hasil eksekusi dari skrip JavaScript!
- Apakah variabel `warna` bernilai "red", "merah", "green", "hijau", "blue", atau "biru"?

Struktur Kondisi: SWITCH

- Struktur **SWITCH** digunakan untuk memberikan beberapa alternatif kondisi untuk eksekusi

```
var warna = "red";
switch (warna) {
    case "green":
        warna = "hijau";
        break;
    case "blue":
        warna = "biru";
        break;
    case "red":
        warna = "merah";
        break;
    default:
        warna = "";
}
warna;
```

- Coba cek hasil eksekusi dari skrip JavaScript!
- Apakah variabel `warna` bernilai “red”, “merah”, “green”, “hijau”, “blue”, atau “biru”?
- Coba bandingkan dengan skrip pada struktur **IF**!

Challenge #3

Demonstrasikan/eksekusi skrip JavaScript yang ada pada *Slide #17* dan *#18*!

Bandingkan hasil dari kedua skrip!

Dengan adanya dua pilihan struktur kondisi, berdasar pertimbangan apakah Anda memutuskan untuk lebih baik menggunakan struktur `IF` atau `SWITCH`?

Struktur Kondisi: FOR

- Digunakan untuk melakukan perulangan jika jumlah perulangan sudah diketahui sebelumnya

```
for (var count = 1; count <= 10; count++)  
{  
    console.log("Iteration number " + count);  
}
```

- Coba cek hasil eksekusi dari skrip JavaScript!
- Berapa kali teks "Iteration number" tercetak?
- Bagaimana hasil variabel `count` tercetak?
- Apa beda hasil perintah `console.log` dan `document.write`?

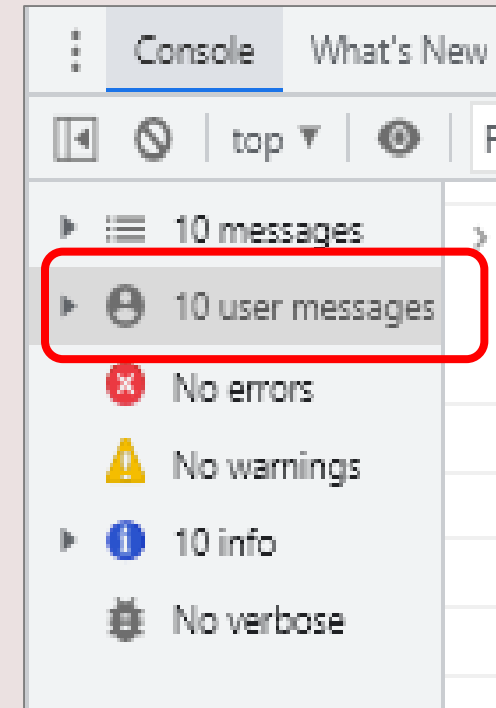
Challenge #4

Demonstrasikan/eksekusi skrip pada
Slide #20!

(Note: jika menggunakan *Web Browser* Chrome, hasil `console.log` dapat dilihat melalui tab “user messages”)

Apa beda hasil perintah `console.log`
dan `document.write`?

Modifikasi skrip JavaScript agar dapat
mencetak teks dengan penomoran
terurut ke bawah pada *console*, yaitu
dari “Iteration number 10” hingga
“Iteration number 1”!



Struktur Perulangan: WHILE

- Digunakan ketika ingin melakukan perulangan selama kondisi yang ditentukan adalah *true*

Struktur WHILE

```
var count = 1;
while (count <= 10)
{
    console.log("Iteration number " + count);
    count++;
}
```

Struktur DO-WHILE

```
var count = 1;
do
{
    console.log("Iteration number " + count);
    count++;
}
while (count <= 10)
```

- Coba cek hasil eksekusi dari skrip JavaScript!
- Berapa kali teks "Iteration number" tercetak?
- Bagaimana hasil variabel `count` tercetak?
- Coba bandingkan dengan skrip pada struktur **FOR**!
- Apa beda antara struktur **WHILE** dan **DO-WHILE**?

Challenge #5

Demonstrasikan/eksekusi skrip pada *Slide* #22!

Modifikasi skrip JavaScript agar dapat mencetak teks dengan penomoran terurut ke bawah pada *console*, yaitu dari “Iteration number 10” hingga “Iteration number 1”!

Challenge #6

Dengan adanya tiga pilihan struktur perulangan, berdasar pertimbangan apakah Anda memutuskan untuk lebih baik menggunakan struktur FOR, WHILE atau DO-WHILE?

Fungsi (*Function*)

- Fungsi adalah sub-program yang dibuat untuk melaksanakan suatu pekerjaan tertentu, seperti menghitung hasil penjumlahan, perkalian, luas suatu area, dll
- Fungsi di dalam Javascript merupakan sebuah *object*, karena ia memiliki *method* (metode) dan *properties*

Deklarasi Fungsi

Memerlukan tiga komponen:

- **Nama fungsi.** Suatu fungsi harus diikuti oleh tanda kurung (“()”) dan aturan penamaan fungsi sama dengan aturan penamaan variabel
- **Nilai atau argumen yang diperlukan** (bila ada). Daftar argumen direpresentasikan melalui parameter yang diletakkan di dalam tanda kurung: (parameter1, parameter2, ...)
- **Operasi yang akan dilakukan di dalam badan fungsi.** Operasi yang akan dieksekusi oleh fungsi diletakkan di dalam tanda kurung kurawal (“{}”). Fungsi yang menghasilkan suatu nilai keluaran harus menggunakan perintah **return**

Contoh Fungsi

```
// deklarasi fungsi
function addNumber(value1, value2) // nama fungsi adalah
"addNumber", memiliki dua parameter (masukan) yaitu "value1"
dan "value2"
{
    total = value1 + value2; // proses yang dilakukan dalam
fungsi adalah melakukan operasi penjumlahan antara nilai
variabel (berdasarkan parameter) "value1" dengan "value2".
Hasil disimpan di dalam variabel "total"

    return total; // keluaran fungsi adalah nilai dari
variabel "total"
}

// memanggil fungsi
sum = addNumber(1, 2); // memanggil fungsi "addNumber".
Parameter "value1" bernilai 1 dan "value2" bernilai 2. Hasil
keluaran fungsi disimpan di dalam variabel "sum"
```

Challenge #7

Demonstrasikan/eksekusi skrip pada *Slide* #27!

Built-in Objects

■ String

- `length`, `charAt()`, `concat()`, `indexOf()`, `replace()`, `substr()`, `split()`, `dst`
- Contoh:

```
str = "this is a string";  
aIndex = str.indexOf('a');  
strEmail = "noor.ifada@trunojoyo.ac.id";  
arrEmail = strEmail.split("@");  
arrName = arrEmail[0].split(".");  
arrDomain = arrEmail[1].split(".");
```

■ Array

- `length`, `concat()`, `sort()`, `reverse()`, `shift()`, `pop()`, `dst`
- Contoh:

```
names = ["Carrie", "Ashley", "Zia", "Bob"];  
sortName = names.sort();
```

Built-in Objects [2]

■ Math

- E, PI, abs(), cos(), pow(), log(), floor(), sqrt(), dst
- Contoh:

```
intRandomInteger = Math.floor(Math.random()*12);
```

■ Date

- getFullYear(), getTime(), getDay(), dst
- Contoh:

```
date = new Date();  
arrDayNames =  
["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",  
 "Friday", "Saturday"];  
strTodayIs = arrDayNames[date.getDay()];
```

Challenge #8

Demonstrasikan/eksekusi skrip pada *Slide* #29 dan #30!

RegExp

- *RegExp* (*Regular Expression*) adalah sebuah *object* yang digunakan untuk mendeskripsikan pola dari suatu (kumpulan) karakter dan **mengecek kesesuaian suatu teks (atau *string*) dengan pola tersebut**
- Terdiri dari dua komponen yaitu *pattern* dan *modifiers*, dengan sintaks penulisan: “/**pattern**/**modifiers**;”

Simbol	Deskripsi <i>Modifier</i>
i	Mengecek kesesuaian yang <i>case-insensitive</i>
g	Mengecek kesesuaian secara keseluruhan
m	Mengecek kesesuaian <i>multiline</i>

Simbol	Deskripsi
Brackets (untuk mencari batasan karakter)	
[abc]	Mencari karakter yang ada di dalam kurung siku
[^abc]	Mencari karakter yang tidak ada di dalam kurung siku
[0-9]	Mencari karakter (angka) yang ada di dalam kurung siku
[^0-9]	Mencari karakter (non angka) yang tidak ada di dalam kurung siku
(red blue green)	Mencari berdasarkan pilihan yang tersedia
Metacharacters (kumpulan karakter yang memiliki makna tertentu)	
.	Mencari karakter tunggal, kecuali new line atau line terminator
\w	Mencari karakter kata
\W	Mencari karakter bukan kata
\d	Mencari sebuah digit angka
\D	Mencari sebuah karakter non angka
\s	Mencari sebuah karakter whitespace
\S	Mencari sebuah karakter bukan whitespace
\b	Pencarian kesesuaian di awal/akhir kata
\B	Pencarian kesesuaian tidak di awal/akhir kata
\0	Mencari karakter NUL
\n	Mencari karakter new line
\f	Mencari karakter form feed
\r	Mencari karakter carriage return
\t	Mencari karakter tabulasi
\v	Mencari karakter tabulasi vertikal
\xxx	Mencari karakter dengan format angka oktal xxx
\xdd	Mencari karakter dengan format angka heksadesimal dd
\uxxxx	Mencari karakter Unicode dengan format angka heksadesimal xxxx
Quantifiers (menunjukkan kuantitas)	
n+	Sesuai dengan string manapun yang memiliki setidaknya satu n
n*	Sesuai dengan string manapun yang memiliki kemunculan n sebanyak nol atau lebih
n?	Sesuai dengan string manapun yang memiliki kemunculan n sebanyak nol atau satu kali
n{X}	Sesuai dengan string manapun yang memiliki urutan n sebanyak X
n{X,Y}	Sesuai dengan string manapun yang memiliki urutan n sebanyak X hingga Y
n{X,}	Sesuai dengan string manapun yang memiliki urutan n setidaknya sebanyak X
n\$	Sesuai dengan string manapun yang diakhiri dengan n
^n	Sesuai dengan string manapun yang awalnya adalah n
?=n	Sesuai dengan string manapun yang diikuti oleh string n
?!n	Sesuai dengan string manapun yang tidak diikuti oleh string n

http://www.w3schools.com/jsref/jsref_obj_regexp.asp

Contoh RegExp Date

RegExp untuk validasi format tanggal dalam bentuk:
yyyy-mm-dd atau **yyyy/mm/dd**

■ Range tahun: **1800 - 2099**

□ Year part: $^(18|19|20)\backslash d\backslash d[-/]$

Bagian awal

Digit 0 sampai 9

Tanda pemisah

□ Month part: $(0[1-9]|1[012])[-/]$

01 – 09

10 – 12

□ Day part: $(0[1-9]|1[2][0-9]|3[01])\$$

01 – 09

10 – 29

30 – 31

```
function verifyDate(strDate) {  
  // Membuat RegExp Date. Perhatikan bahwa karakter berwarna biru adalah komponen  
  // pattern, sedangkan karakter berwarna merah tua adalah komponen modifier  
  var patt=/^(18|19|20)\backslash d\backslash d[-/](0[1-9]|1[012])[-/](0[1-9]|1[2][0-9]|3[01])$/g;  
  return patt.test(strDate);  
}
```

Challenge #9

Demonstrasikan/eksekusi skrip pada *Slide #34*!
Gunakan fungsi “`verifyDate`” untuk mengecek
beberapa tanggal berikut:

1799-12-31

2020/04/02

2099-13-31

Apakah hasil pengecekan dari masing-masing
tanggal di atas?

RegExp dengan Method

- *Method:*

- `exec()`
 - `test()`
 - `compile()`
 - `toString()`

- *Contoh:*

- `str="Dasar Pemrograman Web";`
 - `patt=/Web/g;`
 - `result = patt.test(str);`

Challenge #10

Demonstrasikan/eksekusi skrip pada Slide #36!
Apakah hasil atau nilai variabel “result”?

Coba ubah nilai dari variabel “str” dan eksekusi ulang perintah berikut:

```
result = patt.test(str) ;
```

Apakah hasil atau nilai variabel “result” saat ini?

Video Perkuliahan

- Penjelasan materi kuliah dapat dilihat via YouTube (*recording* Semester Genap 2020/2021):
<https://youtu.be/3j3FuQyeMew>
- *Note: Kerjakan Challenge(s) berdasarkan slide Semester Genap 2022/2023 (bukan berdasarkan slide yang ditampilkan dalam recording Semester Genap 2020/2021)*



there is
ALWAYS
something to
BE GRATEFUL
for



whitewrap.com

**SURROUND
YOURSELF
WITH POSITIVE
PEOPLE WHO
ARE GOING TO
PUSH YOU
TOWARD
GREATNESS.**

PictureQuotes.com