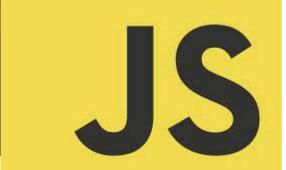
JAVASCRIPT

ASAS PENGATURACARAAN JAVASCRIPT FUNCTION DAN CONTROL FLOW



Function

- Reusable block of code
 - Boleh digunakan semula
 - Memudahkan pemahaman code
 - Memudahkan penyusunan code

```
const nombor5 = 5;
const nombor10 = 10;

function tambahDua(nombor1, nombor2) {
   return nombor1 + nombor2;
}

const hasil = tambahDua(nombor5, nombor10);
console.log(hasil); // 15
```

Anatomi Function

block of code/ statement (with explicit return)

```
parameters

32   const nombor5 = 5;
33   const nombor10 = 10;

34

35   function tambahDua(nombor1, nombor2) {
    return nombor1 + nombor2;
}

37

38

39   const hasil = tambahDua(nombor5, nombor10);
40   console.log(hasil); // 15
```

invoke(call) function tambahDua with 2 arguments

```
const hargaBarang1Usd = 100;
     const hargaBarang2Usd = 200;
     const hargaBarang3Usd = 300;
     const hargaBarang4Usd = 400;
10
11
     const hargaBarang1Myr = hargaBarang1Usd
12
     const hargaBarang2Myr = hargaBarang2Usd * 4.2;
13
     const hargaBarang3Myr = hargaBarang3Usd * 4.2;
     const hargaBarang4Myr = hargaBarang4Usd * 4.2;
14
15
16
     console.log(hargaBarang1Myr);
17
     console.log(hargaBarang2Myr);
     console.log(hargaBarang3Myr);
18
     console.log(hargaBarang4Myr);
19
20
     // function declaration
21
     function convertUsdToMyr(hargaUsd) {
22
23
       return hargaUsd * 4.2;
24
25
     console.log(convertUsdToMyr(hargaBarang1Usd));
26
     console.log(convertUsdToMyr(hargaBarang2Usd));
27
     console.log(convertUsdToMyr(hargaBarang3Usd));
28
29
     console.log(convertUsdToMyr(hargaBarang4Usd));
```

Bentuk-bentuk function

```
function pendarabDua(nombor) {
  return nombor * 2;
}
```

```
const pendarabDua = function(nombor) {
return nombor * 2;

}
```

```
const pendarabDua = (nombor) => {
return nombor * 2;
}
```

Scope

Penting untuk memahami data dan maklumat yang boleh diakses oleh function

```
60   const nama = 'John';
61
62   function panggilNama() {
63      const nama = 'Doe';
64      console.log(nama);
65   }
66
67   panggilNama(); // Doe
68
69   console.log(nama); // John
70
```

Function akan menggunakan variable yg wujud dalam scope sendiri terlebih dahulu

```
const nama = 'John';

const panggilNamaLagi = () => {
    console.log(nama);

panggilNamaLagi(); // John

panggilNamaLagi(); // John
```

Jika tiada dalam scope sendiri, dia akan mencari dalam variable dalam scope luar dia

```
88  Let nama = 'John';
89
90  function panggilNama() {
91    nama += 'Doe';
92    console.log(nama);
93  }
94
95  panggilNama(); // John Doe
96
97  console.log(nama); // John Doe
98
```

Mutation data dalam function akan mutate variable asal

Default parameters

```
function panggilNama(nama = 'John') {
console.log(nama);
}

panggilNama(); // John

panggilNama('Doe'); // Doe
```

Function boleh mempunyai parameters yang value ditetapkan

Control flow

- Cara Javascript susunan code execution
- Conditional and loops
- Conditional
 - ▶ If
 - ► If ... else
 - ► If ... else if
 - Switch
- Loops
 - For
 - Foreach

if

switch

```
const day = 2;
66 v switch (day) {
67 v case 1:
        console.log('Monday');
        break;
70 v case 2:
        console.log('Tuesday');
71
       break:
72
73 v case 3:
        console.log('Wednesday');
74
75
       break;
76 v case 4:
        console.log('Thursday');
        break;
78
79 v case 5:
        console.log('Friday');
       break;
81
82 v case 6:
        console.log('Saturday');
       break;
85 v case 7:
        console.log('Sunday');
87
       break;
88 v default:
        console.log('Invalid day');
        break;
```

Terima Kasih. Soalan?