M02 - JavaScript Fundamentals

•••

Conditionals

Conditionals

- Sometimes, we need to take different actions based on different conditions
- To do that, we use:
    - if - to specify a block of code to be executed, if certain condition is true
    - else - to specify a block of code to be executed, if the same condition is false
    - else...if - to specify a new condition to test, if the first condition is false
    - ? - ternary operator
    - switch - to specify alternative blocks of code to run

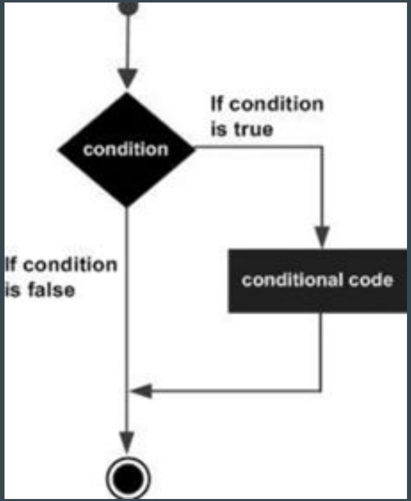
Conditionals

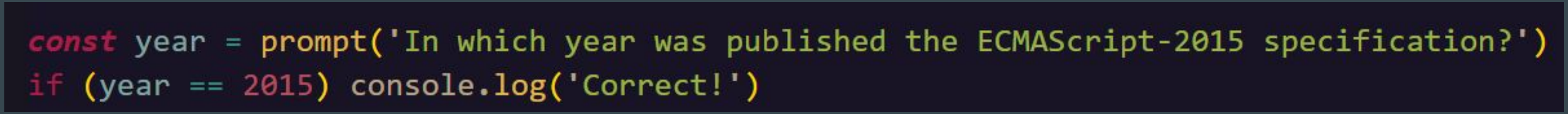
1. IF statement

Conditionals

1. IF statement

- The if statement evaluates a condition and, if the result of the condition is true, executes a block of code



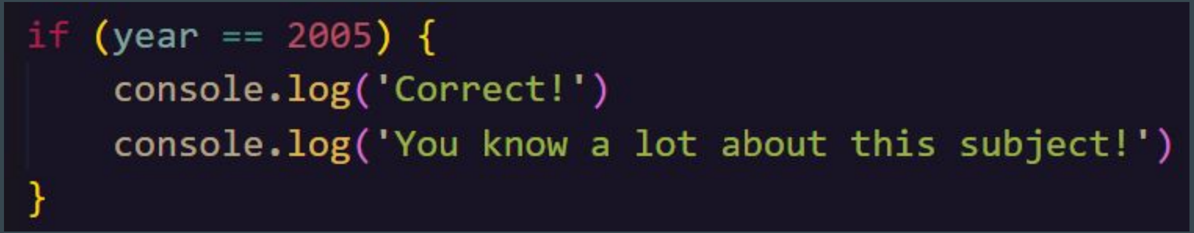


```
const year = prompt('In which year was published the ECMAScript-2015 specification?')
if (year == 2015) console.log('Correct!')
```

Conditionals

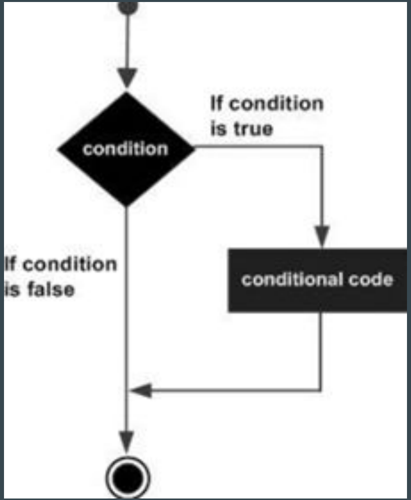
1. IF statement

- If we want to execute more than one declaration, we have to wrap our code block within curly braces:



```
if (year == 2005) {
    console.log('Correct!')
    console.log('You know a lot about this subject!')
}
```

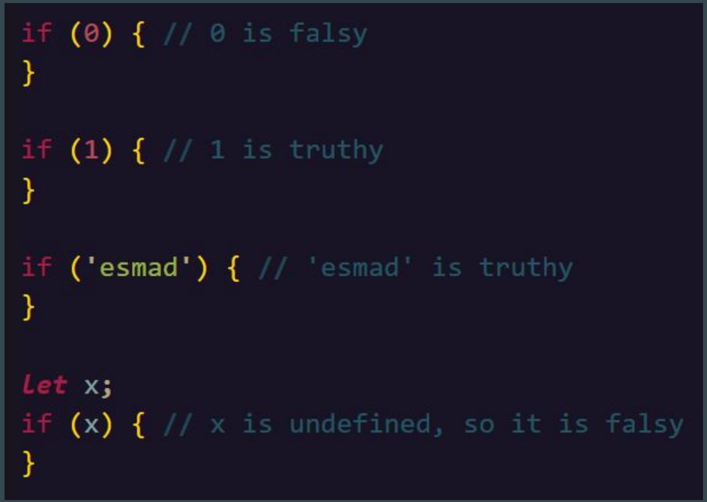
- It is recommended that you always add the block of code with curly braces {}, even if there is only one instruction to be executed. This improves readability!



Conditionals

1. IF statement

- Boolean conversion
  - The if (...) statement evaluates the expression in parentheses and converts the result to a Boolean
  - Rules:
    - i. 0, empty string "", null, undefined and NaN are false. They are called falsy values
    - ii. Any other value becomes true. They are called truthy values



```
if (0) { // 0 is falsy
}

if (1) { // 1 is truthy
}

if ('esmad') { // 'esmad' is truthy
}

let x;
if (x) { // x is undefined, so it is falsy
}
```

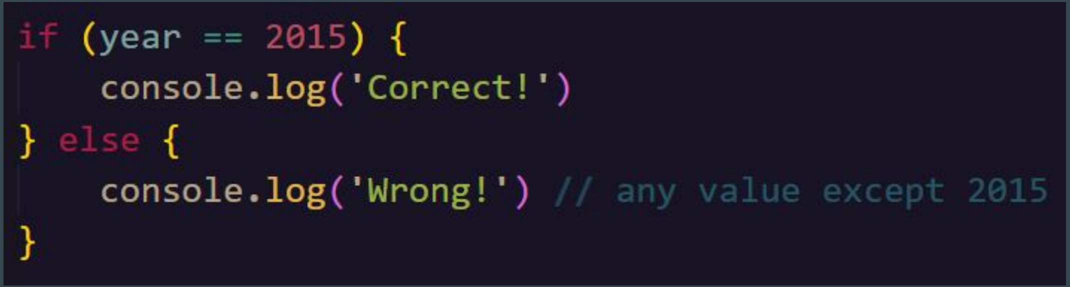
Conditionals

2. ELSE statement

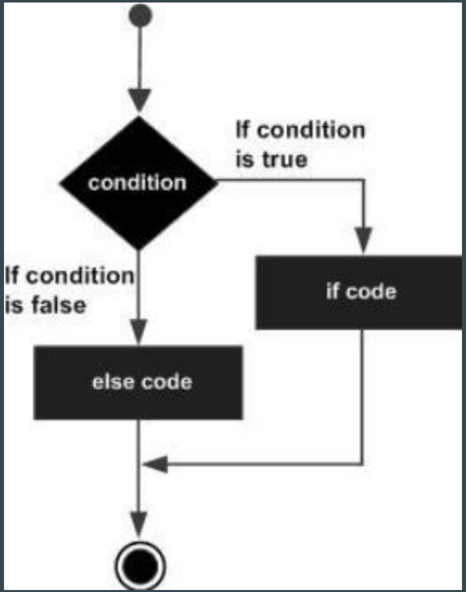
Conditionals

2. ELSE statement

- The if statement can contain an optional else block
- It is executed when the condition is false



```
if (year == 2015) {
    console.log('Correct!')
} else {
    console.log('Wrong!') // any value except 2015
}
```



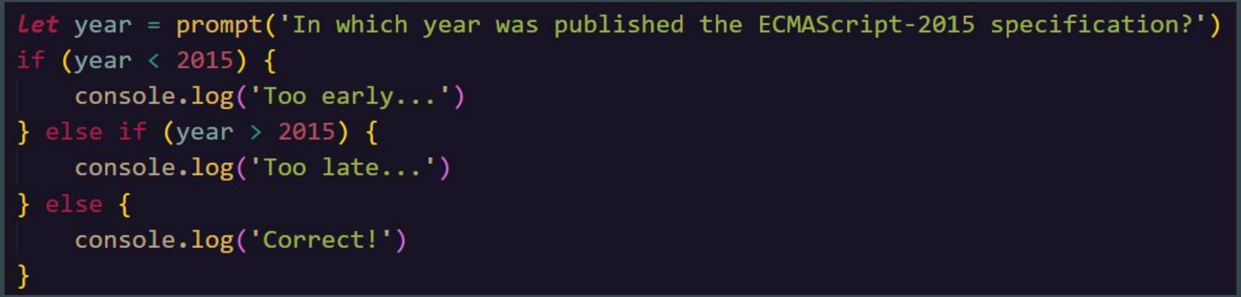
Conditionals

3. ELSE...IF statement

Conditionals

3. ELSE...IF statement

- Sometimes, we would like to test several variants of a condition
- The else if clause allows us to do that



```
let year = prompt('In which year was published the ECMAScript-2015 specification?')
if (year < 2015) {
    console.log('Too early...')
} else if (year > 2015) {
    console.log('Too late...')
} else {
    console.log('Correct!')
}
```

- In the code above, JavaScript first checks the year < 2015. If it is false, it goes to the next year > 2015 condition. If it is also false, it shows the last alert.
- There may be more else if blocks. The final else is optional

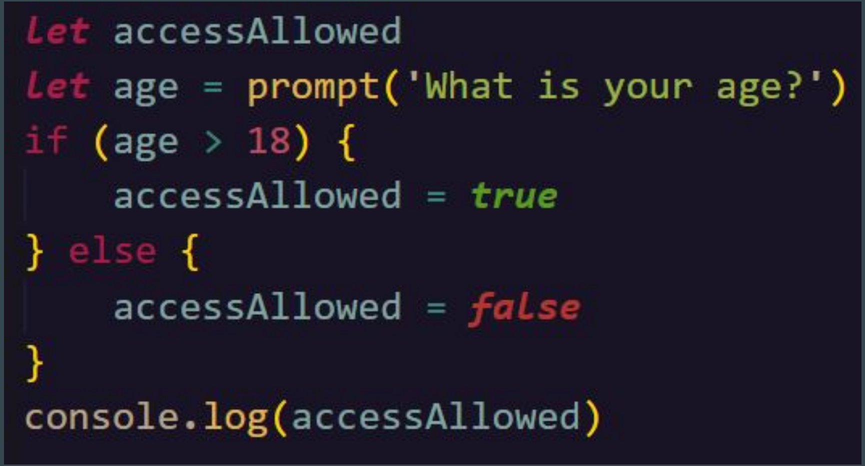
Conditionals

4. ? Ternary Operator

Conditionals

4. Ternary operator ?

- Sometimes, we need to assign a variable depending on a condition

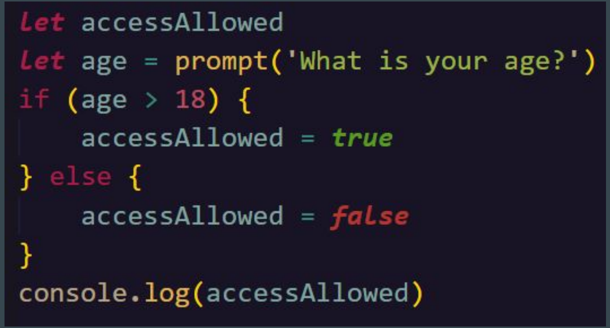


```
let accessAllowed
let age = prompt('What is your age?')
if (age > 18) {
    accessAllowed = true
} else {
    accessAllowed = false
}
console.log(accessAllowed)
```

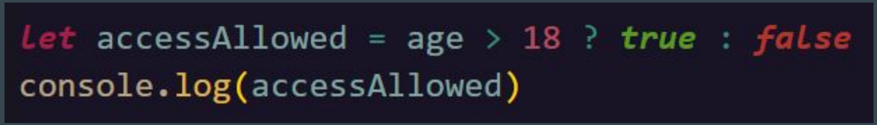
Conditionals

4. Ternary operator ?

- The ternary operator (?) allows us to do this in a shorter and simpler way
- The formal term ternary means that the operator has three operands



```
let accessAllowed
let age = prompt('What is your age?')
if (age > 18) {
    accessAllowed = true
} else {
    accessAllowed = false
}
console.log(accessAllowed)
```



```
let accessAllowed = age > 18 ? true : false
console.log(accessAllowed)
```

- The condition (age>18) is evaluated: if true, the value true is returned and assigned to the variable accessAllowed. If false, the value false is assigned.

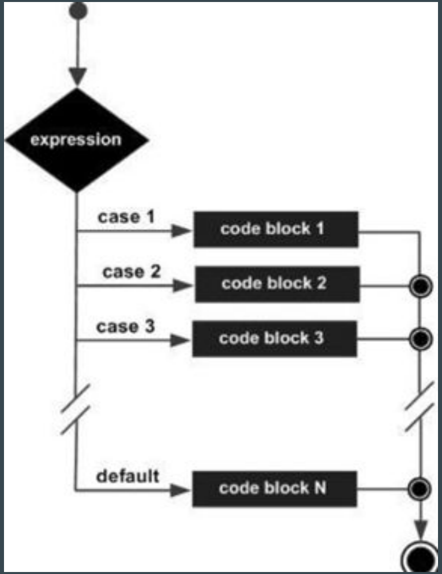
Conditionals

5. SWITCH statement

Conditionals

5. SWITCH statement

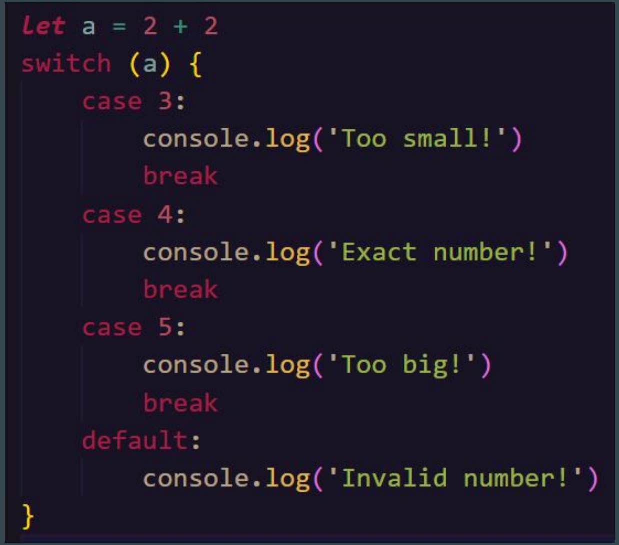
- The switch statement can replace multiple if checks
  - More descriptive way to compare a value with multiple variants
  - Switch has one or more case blocks and an optional default action



Conditionals

5. SWITCH statement

- How does it work?
    - The value of a is checked for strict equality with the value of the first case (that is, 3), then for the second case (4) and so on.
    - If equality is found, the switch starts executing the code from the corresponding case, until the nearest break (or until the end of the switch).
    - If no cases are matched, the default block is executed (if any) If there is no break, execution continues with the next case without any verification!



```
let a = 2 + 2
switch (a) {
    case 3:
        console.log('Too small!')
        break
    case 4:
        console.log('Exact number!')
        break
    case 5:
        console.log('Too big!')
        break
    default:
        console.log('Invalid number!')
}
```