

B I T

Programming Principles

with **JavaScript**

B I T

Conditions

Conditions

Conditions provide a simple but powerful way to **control the flow** of code execution

```
var result = '';  
var a = 3;  
  
if (a > 2) {  
    result = a + ' is greater than 2';  
}  
  
console.log(result);  
// Output: 3 is greater than 2
```

Conditions

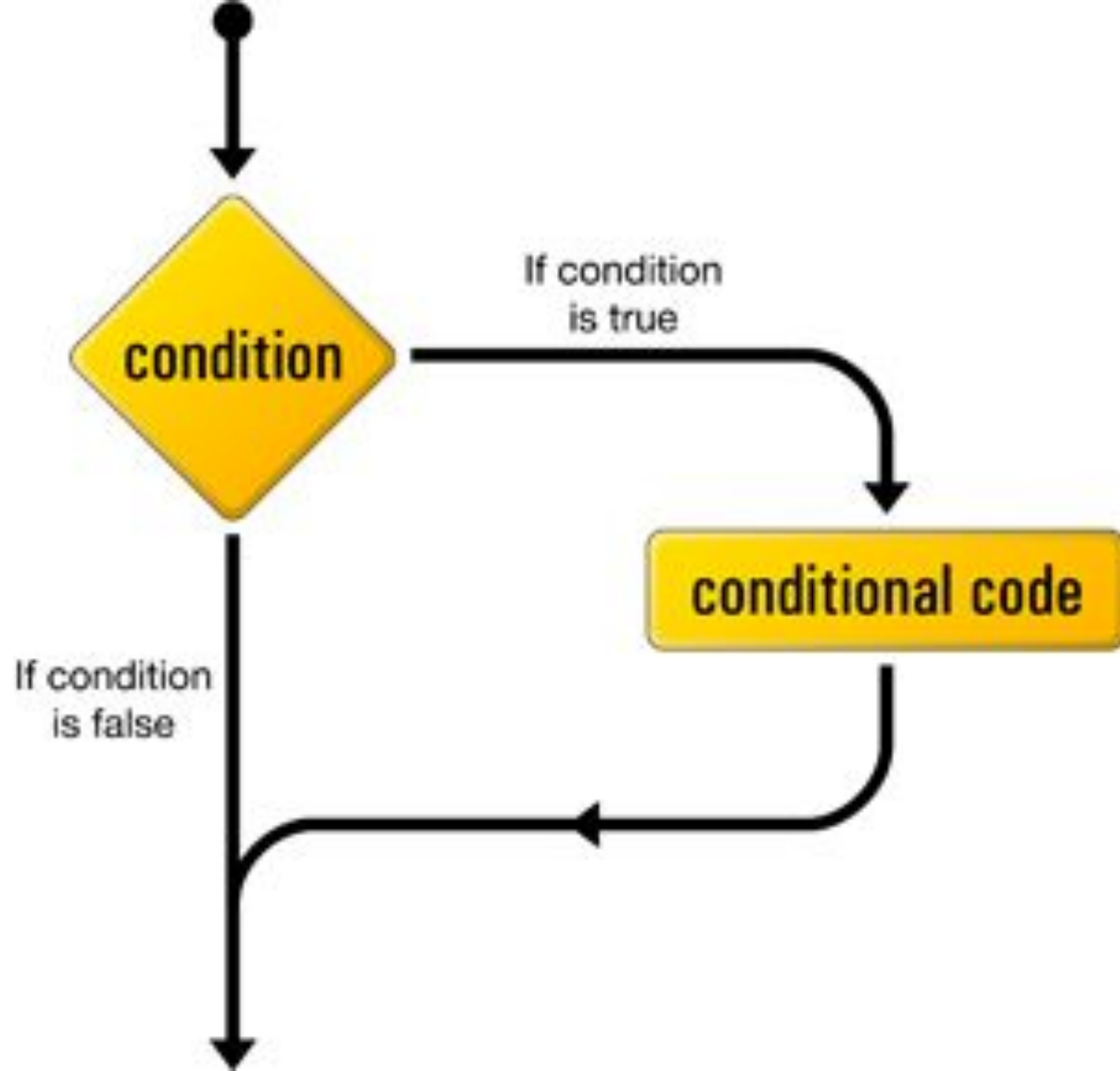
The parts of the if condition are:

- The `if` statement

- A condition in parentheses `()`

- A block of code wrapped in `{ }` that executes if the condition is satisfied

B I T



Conditions

The condition (the part in parentheses) always returns a **Boolean value**

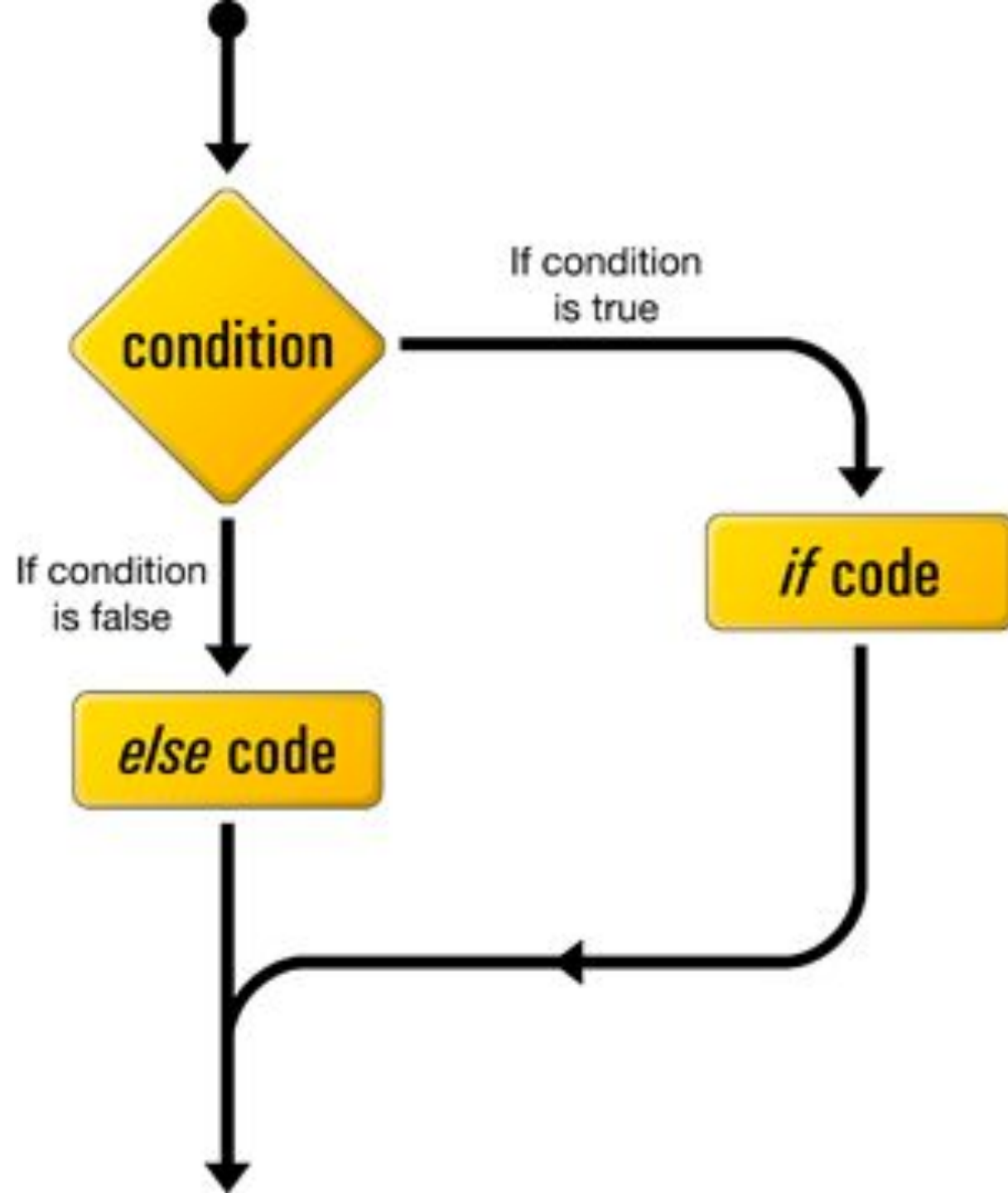
- A logical operation: `!`, `&&`, `or` `||`
- A comparison, such as `===`, `!=`, `>`
- Any value or variable that can be converted to a Boolean
- A combination of the above

Conditions

Optional `else` part of the if condition

The `else` statement is followed by a block of code that runs if the condition evaluates to `false`

B I T



Conditions

```
var result = '';  
var a = 3;  
  
if (a > 2) {  
    result = a + ' is greater than 2';  
} else {  
    result = a + ' is NOT greater than 2';  
}  
  
console.log(result);  
// Output: 3 is greater than 2
```

Conditions

```
var result = '';  
var a = 1;  
  
if (a > 2) {  
    result = a + ' is greater than 2';  
} else {  
    result = a + ' is NOT greater than 2';  
}  
  
console.log(result);  
// Output: 1 is NOT greater than 2
```

Conditions

```
var result = '', a = 0, b = 1;
if (a > 2 || a < -2) {
    result = 'a is not between -2 and 2';
} else if (a === 0 && b === 0) {
    result = 'both a and b are zeros';
} else if (a === b) {
    result = 'a and b are equal';
} else {
    result = 'I give up';
}
console.log(result);
```

Conditions

```
var result = '', a = 0, b = 1;
if (a === 1) {
  if (b === 2) {
    result = 'a is 1 and b is 2';
  } else {
    result = 'a is 1 but b is not 2';
  }
} else {
  result = 'a is not 1, no idea about b';
}
console.log(result);
```

Checking if a Variable Exists

It's often necessary to check whether a variable exists

```
var result = '';  
if (somevar) {  
    result = 'yes';  
}  
// ReferenceError: somevar is not defined
```

Checking if a Variable Exists

A better way to check if a variable is defined is to use unary operator `typeof`

```
var result = "";  
if (typeof somevar !== "undefined") {  
    result = "yes";  
}  
console.log(result);  
// Output: ""
```

`typeof` always returns a string, and you can **compare** this string with the string `"undefined"`

Ternary *Operator*

```
(condition) ? value1 : value2
```

is called a **ternary** operator because it takes three operands. It is usually denoted as **?:**

B I T

Practical

Write a program that compares two numbers and display the larger. Result should be displayed in the console.

Write a program to check if the number is divisible by 2. If it is, print even, if not, print odd.

Sample numbers: 3, 4, 9 (check one at the time)

Output: odd, even, odd

Write a program to check if the number is divisible by 3 and 5.
If it is, print that number.

Sample numbers: **15, 12 (check one at the time)**

Output: **15**

Write a conditional statement to find the sign of product of three numbers. Display the result in the console with the specified sign.

Sample numbers: 3, -7, 2

Output: **The sign is -**

Write a program to check if the variable is a number. If it's a number, check if it is divisible by 2. If it is, print the result, if not, show "X"

Sample numbers: **10** | **7** (check one at the time)

Output: **10 / 2 = 5** | **x**

Write a conditional statement to find the largest of five numbers. Display the result in console.

Sample numbers: -5, -2, -6, 0, -1

Output: 0

Write a conditional statement to sort three numbers.

Sample numbers : 0, -1, 4

Output : 4, 0, -1

B I T

Institute of
Technology
Belgrade