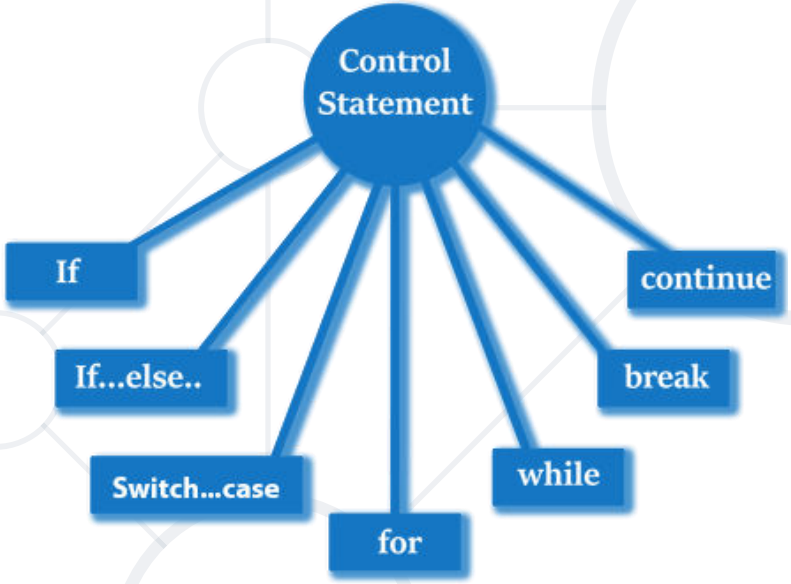
Control-Flow Logic in JS

Operators, Expressions, Statements, Conditional Statements, Loops



SoftUni Team
Technical Trainers

Software University

SoftUni Foundation

Software University
http://softuni.bg

Table of Contents

SoftUni Foundation

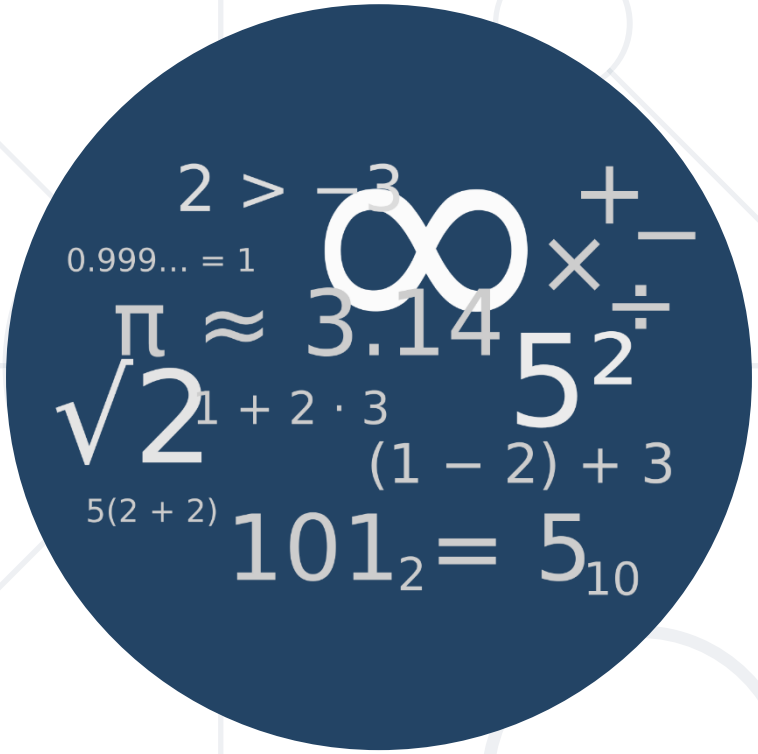
1. Operators, Expressions, Statements
2. Conditional Statements
   - if-else, switch-case
3. Loops
   - for, while, do-while, for-in, for-of

Have a Question?

SoftUni Foundation

sli.do

#JSCORE



Operators and Expressions in JS

Arithmetic, Logical, Comparison, Assignment, Others

Arithmetic Operators in JS

SoftUni Foundation

```
console.log(3 + 4 - 2); // 5 (add / subtract numbers)
console.log(3 * 2); // 6 (multiply numbers)
console.log(2 ** 10); // 1024 (exponential operator **)
console.log(5 / 2); // 2.5 (divide numbers)
console.log(5 / 0); // Infinity (divide by zero)
console.log(Infinity / Infinity); // NaN (wrong division)
console.log(Math.floor(7 / 3)); // 2 (integral division)
console.log(7 % 3); // 1 (remainder of division)
console.log(5.3 % 3); // 2.3 (remainder of division)
let a = 5; console.log(++a); // 6 (prefixed ++)
console.log(a++); // 6 (postfix ++)
```

Problem: Multiply Two Numbers

SoftUni Foundation

- Write a JS function to multiply two numbers

```
function mult(num1, num2) {

  let result = num1 * num2
  return result
}
```

Test this function in the **judge system**

The judge sends the input as **parameters**, specified in the **problem description**

```
mult(4, 2.5) // returns 10
```

Invoke the above function to test it **locally**

Check your solution here: https://judge.softuni.bg/Contests/288

Problem: Boxes and Bottles

- Write a JS function to calculate how many boxes will be needed to fit **n** bottles if each box fits **k** bottles

20 5	➡	4	15 7	➡	3	5 10	➡	1

```
function boxesAndBottles(n, k) {
  console.log(Math.ceil(n / k));
}
```

Check your solution here: https://judge.softuni.bg/Contests/288

Logical Operators in JS

SoftUni Foundation

- The || operators returns the leftmost "**true**" value:

```
let t = false || 0 || '' || 5 || 'hi' || true;
console.log(t); // 5
```

- The && operators returns the leftmost "**false**" value:

```
let val = true && 'yes' && 5 && null && false;
console.log(val); // null
```

Comparison Operators in JS

SoftUni Foundation

- Comparison operators compare values
- ==, !=, <, >, >=, <=, ===, !==
- The == means "equal after type conversion"
- The === means "equal and of the same type"

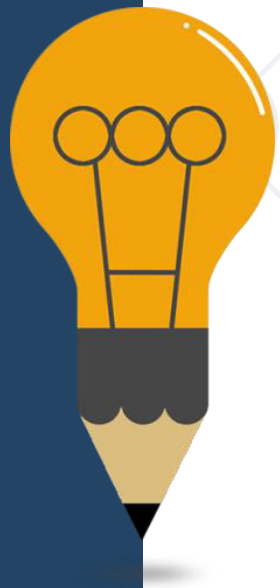
```
var a = 5;
var b = 4;
console.log(a == b); // false
console.log(a != b); // true
console.log(a >= b); // true
console.log(a < "5.5"); // true
console.log(0 == ""); // true
console.log(0 == []); // true
console.log(0 === ""); // false
console.log(3 !== "3"); // true
```

Assignment and Other Operators in JS

SoftUni Foundation

- Assign a value to variable: =, +=, -=, *=, /=, |=, …

```
let y = 4; console.log(y *= 2); // 8
let z = y = 3; // y=3 and z=3
console.log(z += 2); // 5
let unknown_value;
console.log(unknown_value); // undefined
```



- Conditional ternary operator ? :

```
console.log((new Date()).getDay() % 2 === 0 ?
  "even date" : "odd date")
```

Problem: Leap Year

- Write a JS function to check whether a year is leap
  - Can be divided to 4 and cannot be divided to 100
  - Or can be divided to 400

1980	➡	yes	1900	➡	no

```
function leapYear(year) {
  let leap = (year % 4 === 0 && year % 100 !== 0)
      || (year % 400 === 0);
  console.log(leap ? "yes" : "no");
}
```

Check your solution here: https://judge.softuni.bg/Contests/288

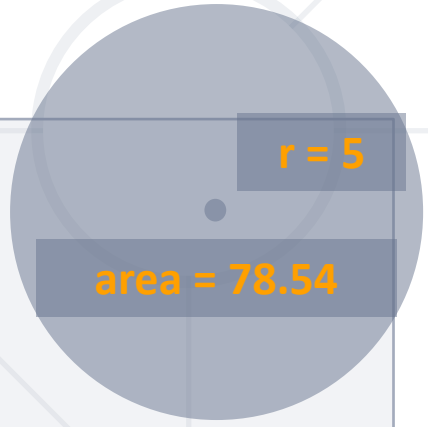
Expressions in JS: Circle Area

- Expressions combine variables, values, operators, function calls
  - Example: calculate circle area by given radius

```
function circleArea(r) {
  let area = Math.PI * r * r;
  console.log(area); // 78.53981633974483

  let areaRounded = Math.round(area * 100) / 100;
  console.log(areaRounded); // 78.54
}
```

Circle area expression




```
circleArea(5);
```

Check your solution here: https://judge.softuni.bg/Contests/288

Expressions in JS: Triangle Area

SoftUni Foundation

- Calculate triangle area by its 3 sides

```
function triangleArea(a, b, c) {
  let sp = (a + b + c) / 2;
  let area = Math.sqrt(sp * (sp - a) * (sp - b) * (sp - c));
  return area;
}
```

Heron's formula

```
triangleArea(2, 3.5, 4);
// 3.4994419197923547
```

Check your solution here: https://judge.softuni.bg/Contests/288

Problem: Cone Volume and Surface Area

- Write a JS function to calculate cone volume and surface
  - The cone height **h** and radius of the base **r** are given

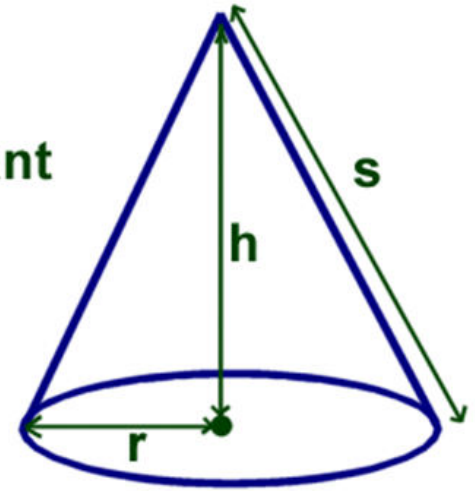
```
3
5
```

```
volume = 47.1239
area = 83.2298
```

```
3.3
7.8
```

```
volume = 88.9511
area = 122.0159
```

r = radius
h = height
s = length of slant



Formulas + online calculator:
http://www.calculatorsoup.com/calculators/geometry-solids/cone.php

Solution: Cone Volume and Surface Area

- Slant height
  - $s = \sqrt{(r^2 + h^2)}$
- Volume
  - $V = \pi * r^2 * h / 3$
- Base surface area
  - $B = \pi * r^2$
- Lateral surface area
  - $L = \pi * r * s$

```
function cone(r, h) {
  let s = Math.sqrt(r * r + h * h);
  let volume = Math.PI * r * r * h / 3;
  console.log("volume = " + volume);
  let area = Math.PI * r * (r + s);
  console.log("area = " + area);
}
```

- Total surface area: $A = B + L = \pi * r * (r + s)$

Check your solution here: https://judge.softuni.bg/Contests/288

Statements in JS

- **Statements** are "**commands**" to be executed

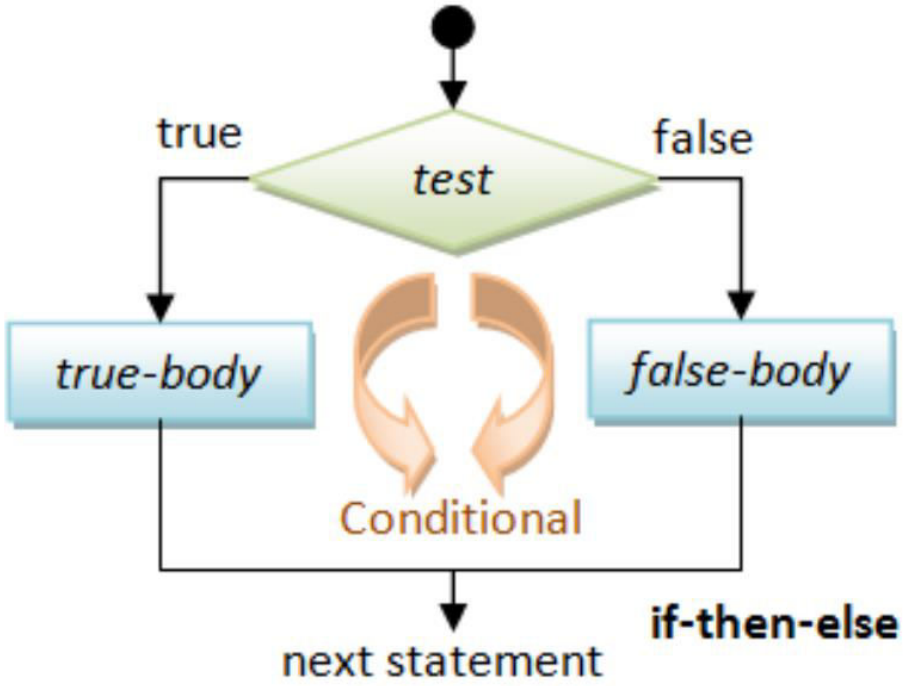
```
let number = 5;
console.log(number)

if (number === 5) {
  number = number + 1
  console.log(number)
}
;
```

Semicolon ; at the end of line is not mandatory in JS

Block { } statements hold a sequence of commands

Empty statement



Conditional Statements

If-else, switch-case

Conditional Statements: if-else

- JS supports the classical **if** / **if-else** statements:

```
let number = 5;

if (number % 2 === 0) {
  console.log("Even number");
} else {
  console.log("Odd number");
}
```

Problem: Odd / Even

- Write a JS function to check if a number is **odd** or **even** or **invalid**

```
function oddEven(num) {
  if (!Number.isInteger(num))
    console.log("invalid")
  else if (num % 2 === 0)
    console.log("even")
  else
    console.log("odd")
}
```

5	→	odd
8	→	even
-3	→	odd
1.5	→	invalid
0	→	even

Check your solution here: https://judge.softuni.bg/Contests/288

Truthy and Falsy Expressions in JavaScript

SoftUni Foundation

- JavaScript is rich of unexpected behaviour

```
console.log("0" == true) // false
console.log("0" == false) // true
if ("0") console.log(true) // true

console.log([] == true) // false
console.log([] == false) // true
if ([]) console.log(true); // true

console.log(null == false || null == true) // false
if (!null) console.log(true); // true
```

The switch-case Statement

- Selects for execution a statement from a list depending on the value of the **switch** expression

```
let day = 3
switch (day) {
    case 1: console.log('Monday'); break;
    case 2: console.log('Tuesday'); break;
    case 3: console.log('Wednesday'); break;
    …
    case 7: console.log('Sunday'); break;
    default: console.log('Error!'); break;
}
```

Problem: Fruit or Vegetable

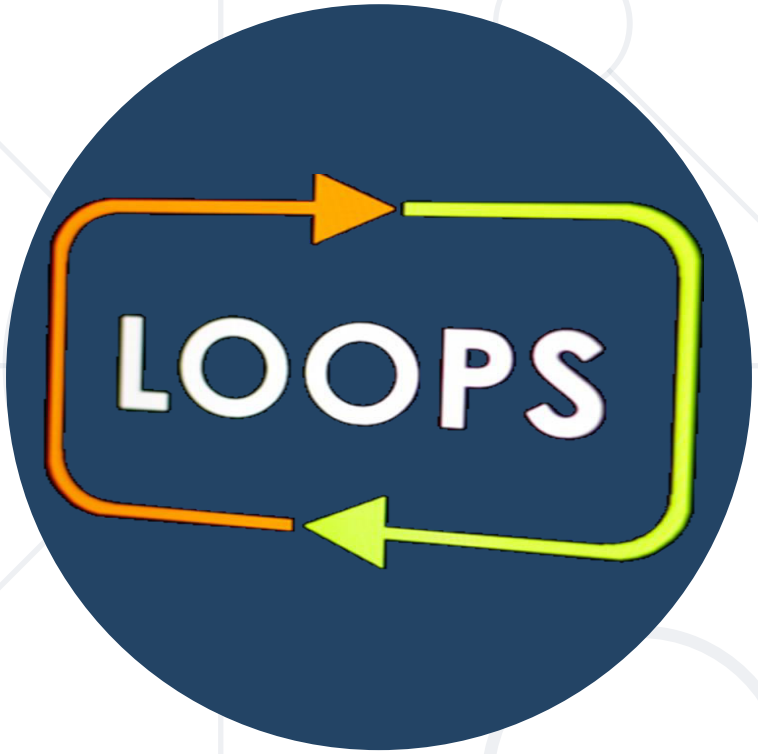
- Write a JS function to print "**fruit**", "**vegetable**" or "**unknown**" depending on the input string
  - Fruits are: banana, apple, kiwi, cherry, lemon, grapes, peach
  - Vegetable are: tomato, cucumber, pepper, onion, garlic, parsley
  - All others are unknown

`lemon` →	`fruit`	→ `peach` →	`fruit`
`onion` →	`vegetable`	→ `pizza` →	`unknown`

Solution: Fruit or Vegetable

```
function food(word) {
  switch (word) {
    case 'banana':
    case 'apple':
    case 'kiwi':
    case 'cherry':
    case 'lemon':
    case 'grapes':
    case 'peach':
      console.log('fruit');
      break;
    case 'tomato':
    case 'cucumber':
    case 'pepper':
    case 'onion':
    case 'parsley':
    case 'garlic':
      console.log('vegetable');
      break;
    default:
      console.log('unknown');
  }
}
```

Check your solution here: https://judge.softuni.bg/Contests/288



Loops in JS

For, while, do-while

Loops: for

- The **for** / **while** / **do-while** loops work as in C++, C# and Java
- Classical **for**-loop

```
for (let i = 0; i <= 5; i++)
  console.log(i)
// 0 1 2 3 4 5
```

```
for (let i = 50; i >= 10; i -= 10)
  console.log(i)
// 50 40 30 20 10
```

Loops: while, do-while, …

```
let count = 1
while (count < 1024)
  console.log(count *= 2)
// 2 4 8 16 32 64 128 256 512 1024
```

```
let s = "ho"
do {
  console.log(s);
  s = s + s;
}
while (s.length < 20)
// ho hoho hohohoho hohohohohohohoho
```

Problem: Colorful Numbers 1 … n

- Write a JS function to print the numbers from 1 to n
  - Return a string holding HTML list **
…
**
  - Display the odd lines in **green**, even lines in **blue**

```
<ul>
  <li><span style='color:green'>1</span></li>
  <li><span style='color:blue'>2</span></li>
  <li><span style='color:green'>3</span></li>
  …
</ul>
```

Solution: Colorful Numbers 1 … n

```
function nums(n) {
  let html = '<ul>\n';
  for (let i = 1; i <= n; i++) {
    let color = 'blue';
    if (i % 2 !== 0) color = 'green';
    html += `<li><span style='color:${color}'>${i}</span></li>\n`;
  }
  html += '</ul>';
  return html;
}
```

```
document.body.innerHTML = nums(10)
```

Check your solution here: https://judge.softuni.bg/Contests/288

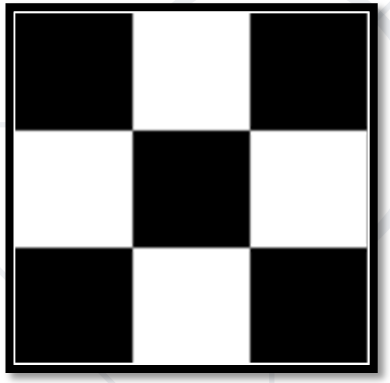
Live Exercises

Problem: Chessboard

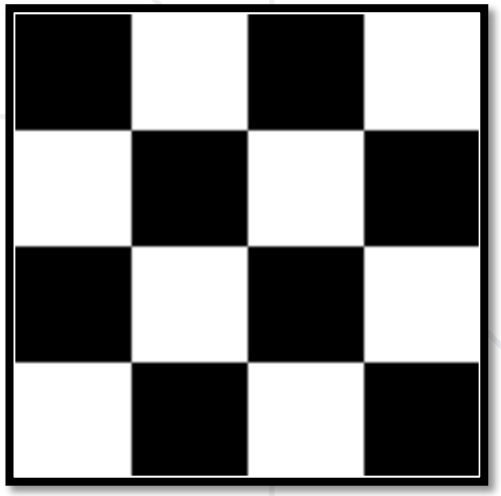
- Write a JS function to print a chessboard of size **n**. Examples:

```
<div class="chessboard">
  <div>
    <span class="black"></span>
    <span class="white"></span>
  </div>
  <div>
    <span class="white"></span>
    <span class="black"></span>
  </div>
  <div>…</div>
</div>
```

n = 3



n = 4



Solution: Chessboard

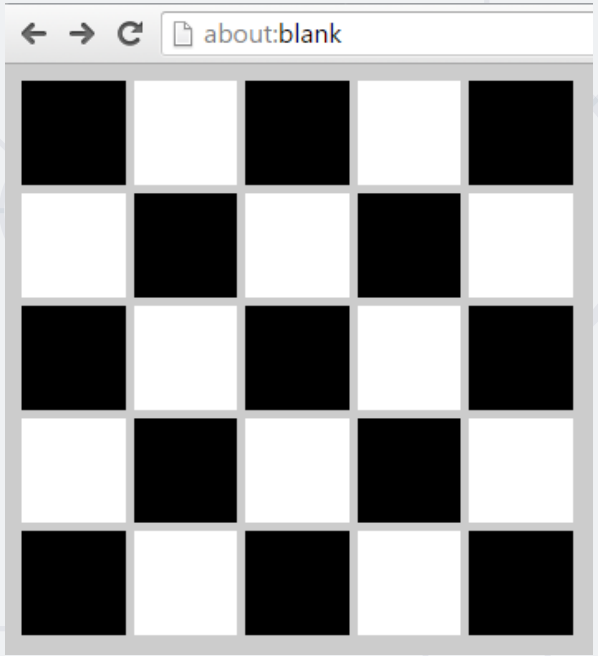
```
function chessboard(size) {
  let html = '<div class="chessboard">\n';
  for (let row = 0; row < size; row++) {
    html += '  <div>\n';
    let color = (row % 2 === 0) ? 'black' : 'white';
    for (let col = 0; col < size; col++) {
      html += `    <span class="${color}"></span>\n`;
      color = (color === 'white') ? 'black' : 'white';
    }
    html += '  </div>\n';
  }
  return html + '</div>';
}
```

Check your solution here: https://judge.softuni.bg/Contests/288

Chessboard: Visualization in the Browser

```
let css = document.createElement("style");
css.innerHTML = `
  body { background: #CCC; }
  .chessboard { display: inline-block; }
  .black, .white {
     width:50px; height:50px;
     display: inline-block; }
  .black { background: black; }
  .white { background: white; }
`;
document.getElementsByTagName("head")[0].appendChild(css);

document.body.innerHTML = chessboard(5);
```



For Each Loop

- for … in loop

```
let nums = [5, 10, 15, 20, 'maria', true];
for (let index in nums)
  console.log(index);
// 0 1 2 3 4 5 → loops through the indices (keys), not values
```

- for … of loop

```
let nums = [5, 10, 15, 20, 'maria', true];
for (let value of nums)
  console.log(value);
// 5 10 15 20 maria true → loops through the values
```

Problem: Binary Logarithm

- Write a JS function to enter **n** numbers and print for each number **x** its binary logarithm ($\log_2 x$)

```
function binaryLogarithm(nums) {
  for (let x of nums) {
    console.log(Math.log2(x));
  }
}
```

```
1024
1048576
256
1
2
```

```
10
20
8
0
1
```

Check your solution here: https://judge.softuni.bg/Contests/288

Problem: Prime Number Checker

```
function isPrime(num) {
  let prime = true;
  for (let d = 2; d < Math.sqrt(num); d++) {
    if (num % d === 0) {
      prime = false;
      break;
    }
  }
  return prime && (num > 1);
}
```

- **break** exits the innermost loop

Check your solution here: https://judge.softuni.bg/Contests/288

Continue

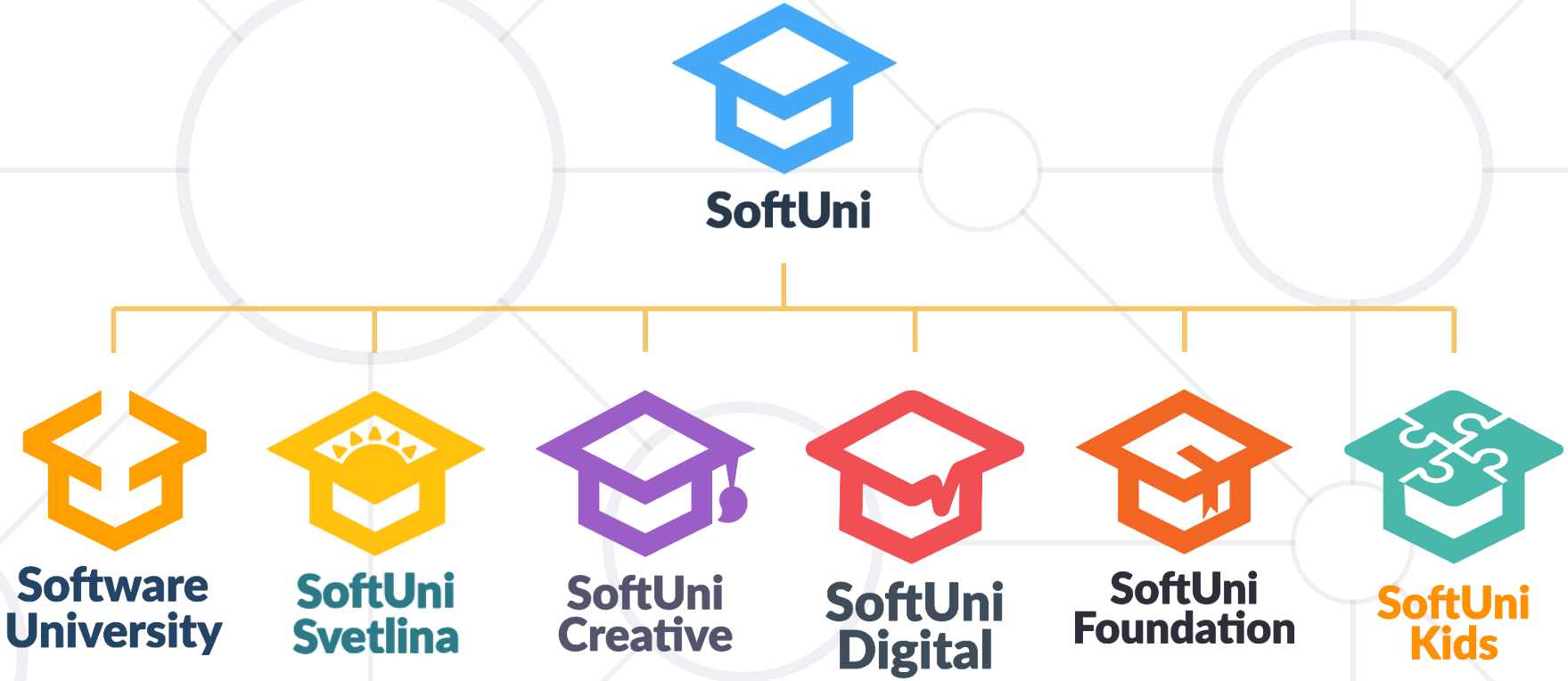
- **continue** goes to the next loop iteration
  - Skips the lines to the end of loop body

```
for (let x = 0; x < 10; x++) {
  if (x % 2 === 0)
    continue;
  console.log(x++);
}
// Prints 1 3 5 7 9
```

Summary

- Operators and expressions in JS are similar to C# / Java / PHP / C++, but not identical
  - **||** returns the leftmost "**true**" expression
- Conditional statements in JS are like in all modern programming languages:
  - Classical conditionals: **if-else**, **switch-case**
- Loops in JavaScript
  - Classical loops: **while**, **do-while**, **for** loops
  - Iterate over collection: **for** … **in** and **for** … **of**

Questions?



https://softuni.bg/trainings/2080/js-fundamentals-september-2018

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech we know sports

telenor

SoftwareGroup doing it right

NETPEAK

SmartIT

Postbank Решения за твоето утре

SUPER HOSTING .BG

INDEAVR Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

codexio

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

WORLD OF MYTHS

Lukanet.com

codexio

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license