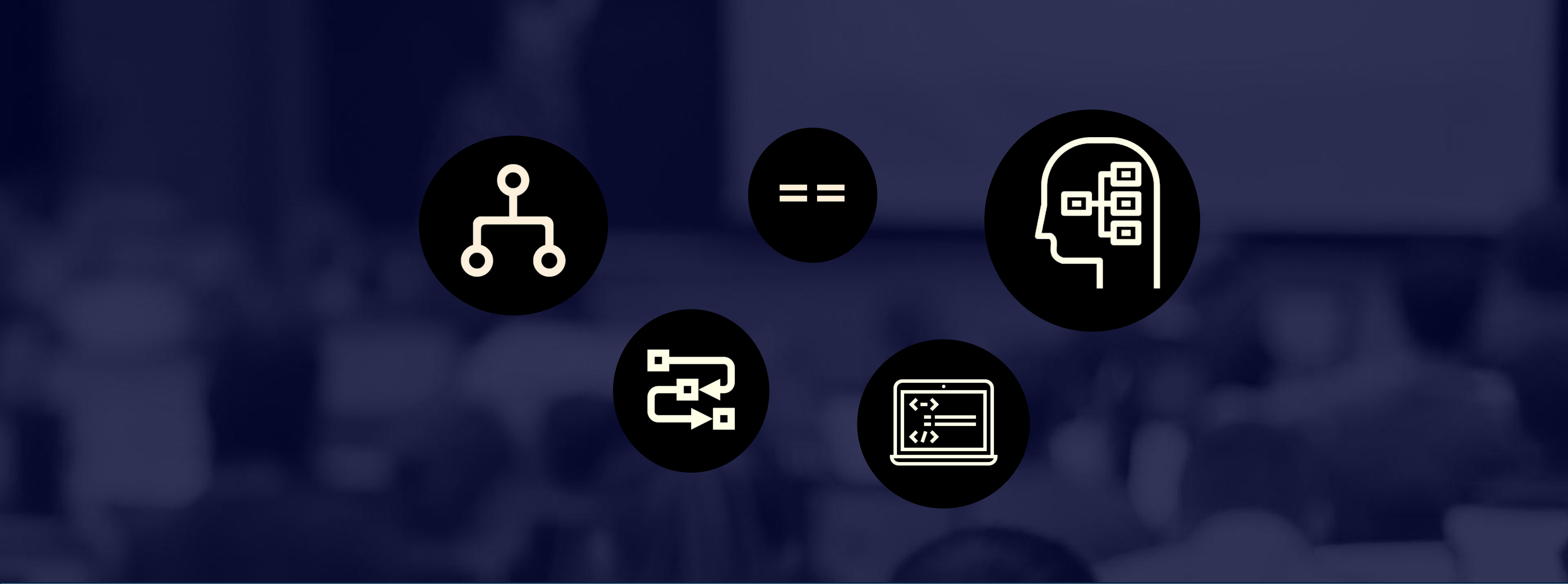
KINGSLAND UNIVERSITY

Conditional Statements



The **if-else** Control-Flow Construction

Table of Contents

- **Review** of the Previous Lesson
- Introduction to **Conditional Statements**
- The **`if`** Statement
- The **`if-else`** Statement
- The **`if-else-if-else-…`** Statement
- **Debugging** in Visual Studio Code
- Practical **Problem Solving**

Review

Expressions and Statements

Variables

- In programming **data** is stored in **variables**
  - **Variables** store data in a piece of the memory
  - Variable **read**: retrieve the stored data
  - Variable **write**: modify the stored data
- Declaring, initializing, reading and changing a variable:

```
let age = 5;
```

```
age = age + 1;
```

```
console.log(age);
```

Data Types in JavaScript

- Simple data types in JavaScript:
  - Number – `2`, `3.14`, `-1`, `1.5e38`, …
  - String – `'hello'`, `'I like JS'`, `"another string"`, …
  - Boolean – `true` or `false`
  - Undefined / Null – missing variable / value

```
console.log(typeof(5)); // number
console.log(typeof('hi')); // string
```

Expressions

- **Expressions** == **variables** and **values**, combined with **operators**

2 is a literal value expression

b * 2 is an arithmetic expression

```
a = b * 2;
```

b is a variable expression

a = b * 2 is an assignment expression

Statements

- **Statements == commands / actions** to be executed

Get the current value stored in b

```
a = b * 2;
```

Multiply that value by 2

Store the result back into another variable we call a

Conditional Statements

Introduction

Real Life Example: Watering Plants

- **If** it is raining:
  - I shall skip watering the plants in the garden
- **Else**:
  - I will have to water them

```
if humidity > 90%
  skip watering
else
  water the plants
```

==

Logical Expressions

Comparison Operators

Comparison Operators

Operators	Designation
Equal value (and type)	== (===)
Not equal value (and type)	!= (!==)
Greater than	>
Greater than or equal to	>=
Less than	<
Less than or equal to	<=

Value Comparison

- In programming we can **compare** values
  - The result of the logical expressions is either `true` or `false`

```
let a = 5;
let b = 10;
console.log(a < b);        // true
console.log(a > 0);        // true
console.log(a > 100);      // false
console.log(a == '5');     // true
console.log(b == 2 * a);   // true
console.log("2" === 2);    // false
```

Conditional Statements

Simple Conditions

Simple Conditions

- Check certain **condition** and take action according to the result

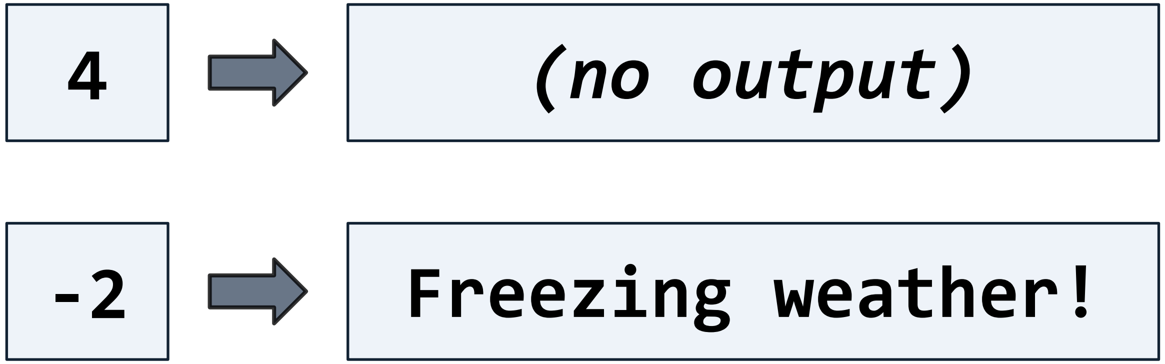
Boolean expression

```
if (condition) {
  // Code for execution if
  // the condition is true
}
```

- The condition evaluates to either `true` or `false`

Problem: Freezing Weather

- Write a function to **check for freezing whether**, which:
  - Receives a temperature in Celsius
  - **Checks** whether the temperature is **below** zero
  - Prints "`Freezing weather!`", if the temperature is equal or smaller than 0



4 → *(no output)*

-2 → Freezing weather!

Solution: Freezing Weather

```
function freezingWeather(temperature) {
   if (temperature <= 0) {
      console.log("Freezing weather!");
   }
}
```

```
freezingWeather(5);
freezingWeather(-0.5); // Freezing weather
```

Simple Conditions: if-else

- If the condition is **`false`**, we may execute another code using the **`else`** block

```
if (condition) {
   // Condition is true
} else {
   // Condition is false
}
```

Blocks of Code

- The curly brackets **`{}`** introduce a **block** (a group of commands)
- In case the **`if`** statement does **not** have curly brackets, only the code on the **next line** will be executed

```
let color = "red";
if (color == "red")
  console.log("tomato");
else
  console.log("banana");
console.log("lemon");
```

Always executed

Blocks of Code

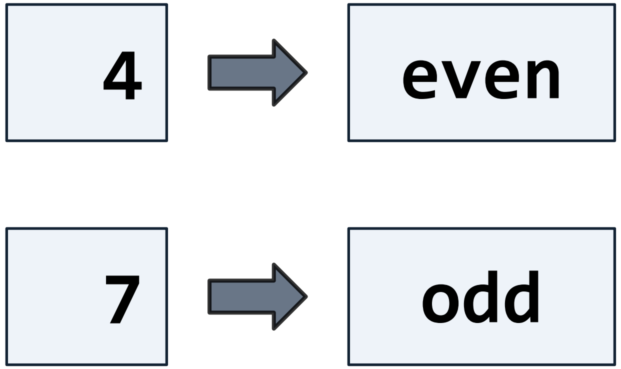
```
let color = "red";
if (color === "red") {
   console.log("tomato");
   console.log("strawberry");
}
else {
   console.log("banana");
   console.log("lemon");
   console.log("pear");
}
```

Block of 2 commands

Block of 3 commands

Problem: Even or Odd

- Write a function to **check for odd or even**, which:
  - Receives a single number
  - If it's even, print "**even**"
  - If it's odd, print "**odd**"



4 → even

7 → odd

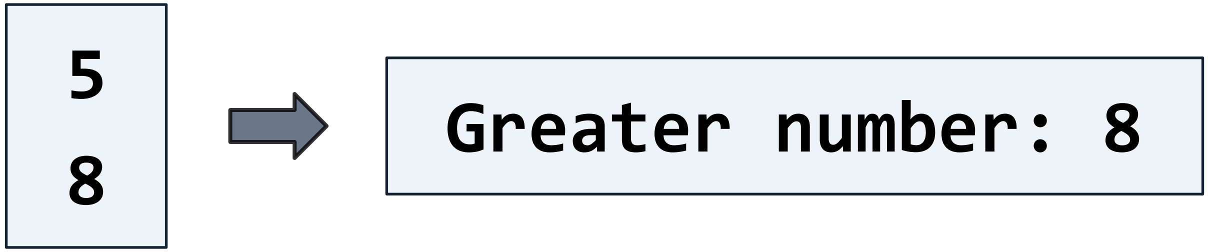
Solution: Even or Odd

```
function evenOrOdd(num) {
   if (num % 2 === 0) {
      console.log("even");
   } else {
      console.log("odd");
   }
}
```

```
evenOrOdd(5);
```

Problem: Greater Number

- Write a function to **find the greater of two numbers**, which:
  - Receives two **numbers**
  - Finds the greater number
  - Prints "**`Greater number: `**" + the **greater** number



```
5
8
```

```
Greater number: 8
```

Solution: Greater Number

```
function greaterNumber(num1, num2) {
  if (num1 > num2) {
    console.log("Greater number: " + num1);
  } else {
    console.log("Greater number: " + num2);
  }
}
```

```
greaterNumber(35, 20);
```

Series of Checks

Complex Conditional Statements

Series of Conditions

- The **`if-else`** statement can be in a series

```
if (…)
  // code
else if (…)
  // code
else if (…)
  // code
```

If one condition is true, the program will NOT check the rest of the conditions

Series of Conditions – Example

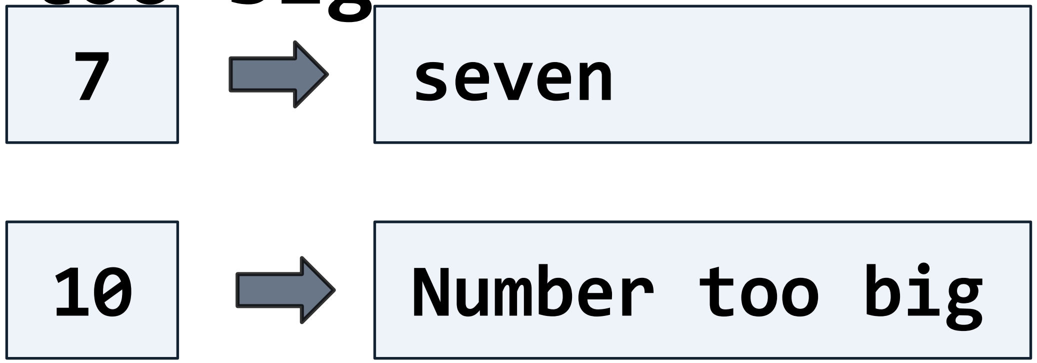
- The program checks the first condition, finds that it is **true** and ends

```
let a = 7;
if (a > 4)
  console.log("Bigger than 4");
else if (a > 5)
  console.log("Bigger than 5");
else
  console.log("Smaller or equal to 4");
```

The output is only "Bigger than 4"

Problem: Number 1…9

- Write a function which **prints a number as text**, which:
  - Receives a **number** in the range [1 … 100]
  - Prints its value in the form of **text** (in English)
  - If the number is **greater** than 9 prints "**Number too big**"



7 → seven

10 → Number too big

Solution: Number 1…9

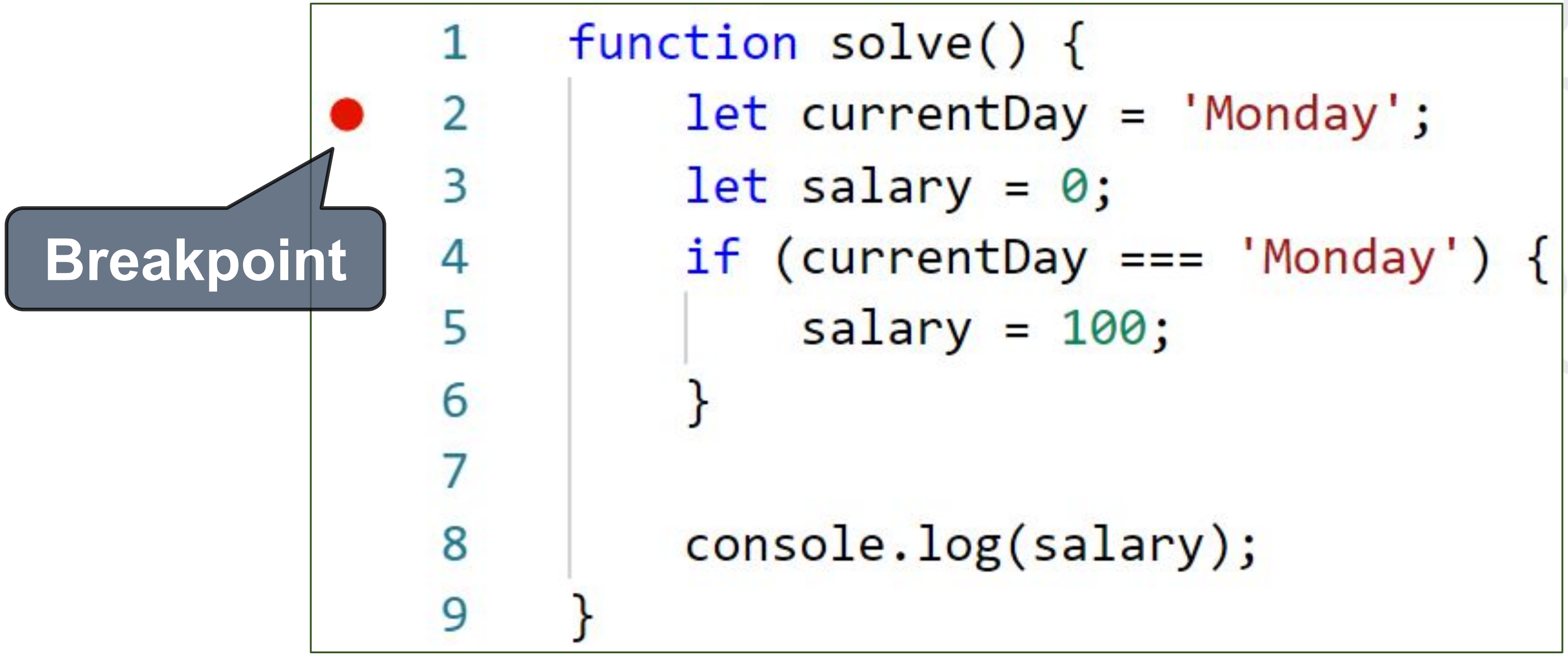
```
function printNumberValue(num) {
   if (num == 1) {
      console.log("one");
   } else if (num === 2) {
      console.log("two");
   }
   // TODO: Add the rest of the checks
   else {
      console.log("Number too big");
   }
}
```

Debugging the Code

Using the Debugger in Visual Studio Code

Debugging

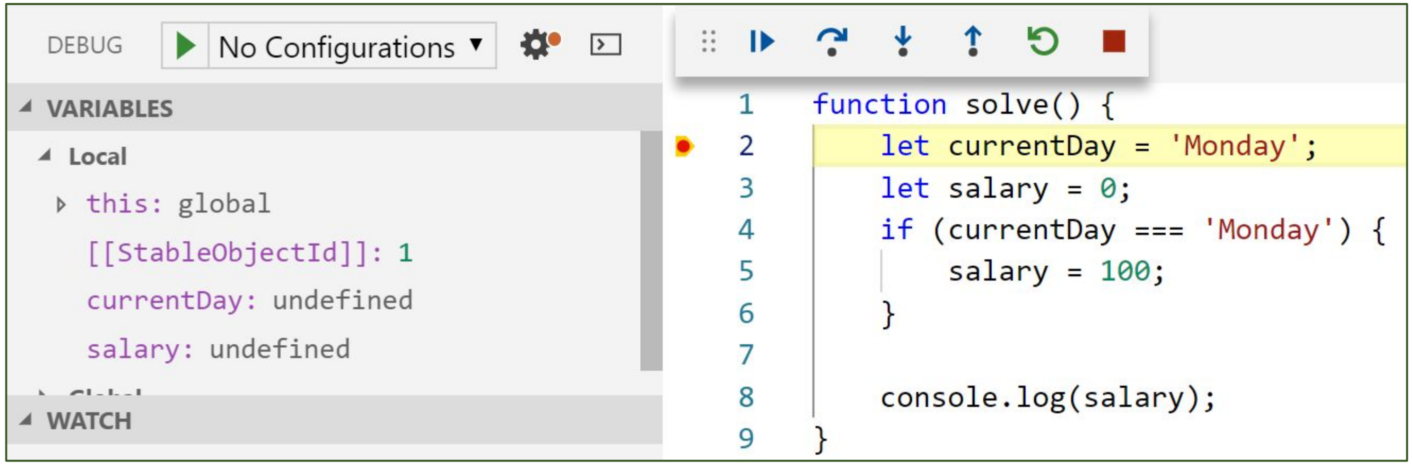
- The process of **tracking** the program execution
  - Debugging allows finding defects (**bugs**)



```
function solve() {
    let currentDay = 'Monday';
    let salary = 0;
    if (currentDay === 'Monday') {
        salary = 100;
    }

    console.log(salary);
}
```

Debugging in Visual Studio Code

- Start the program in **debug mode**: press **[F5]**
- Go to the next step: press **[F10]**
- Add / remove **breakpoint**: press **[F9]**



```
this: global
[[StableObjectId]]: 1
currentDay: undefined
salary: undefined
```

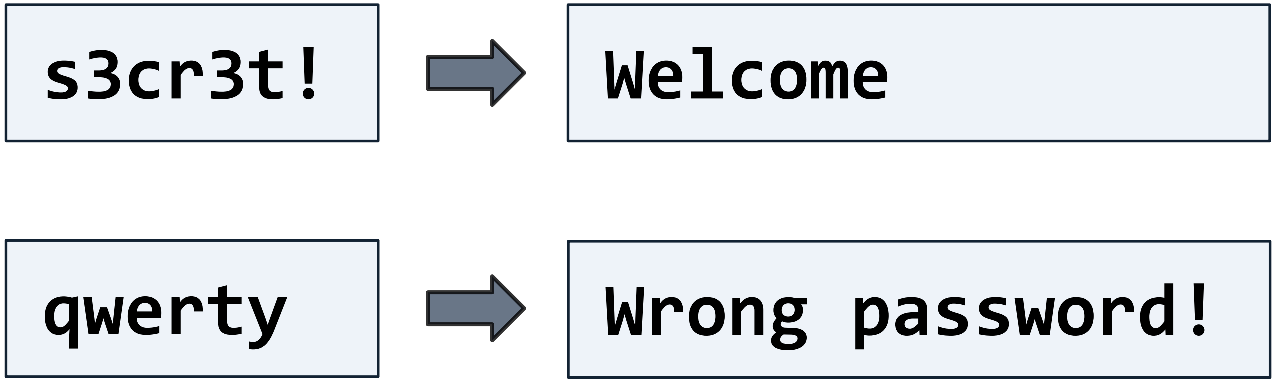
```
function solve() {
    let currentDay = 'Monday';
    let salary = 0;
    if (currentDay === 'Monday') {
        salary = 100;
    }

    console.log(salary);
}
```

Problem Solving

Practical Coding Exercises

Problem: Guess the Password

- Write a function for **checking a password**, which:
  - Receives a string that holds a **password**
  - Prints "**`Welcome`**" if the password is "**`s3cr3t!`**"
  - Prints "**`Wrong password!`**" in all other cases



```
s3cr3t!
```

→

```
Welcome
```

```
qwerty
```

→

```
Wrong password!
```

Solution: Guess the Password

```
function guessThePassword(password) {
  if (password == "s3cr3t!") {
    console.log("Welcome");
  } else {
    console.log("Wrong password!");
  }
}
```

```
guessThePassword("wrong!pass);
```

Problem: Boiling Water

- Write a function to **check for boiling water**, which:
  - Receives a **number:** the water temperature (in °C)
  - Prints "**`The water is boiling`**" if the **number > 100**
  - Prints "**`The water is not hot enough`**" in all other cases

`104.8`	→	`The water is boiling`
`29`	→	`The water is not hot enough`

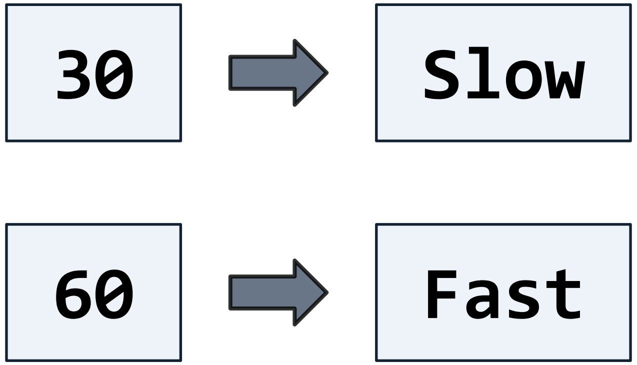
Solution: Boiling Water

```
function boilingWater(degrees) {
    if (degrees > 100) {
        console.log("The water is boiling");
    } else {
        console.log("The water is not hot enough");
    }
}
```

```
boilingWater(108.5);
```

Problem: Speed Info

- Write a function to **check for fast / slow speed**, which:
  - Receives a number (**`speed`**)
  - Prints "**`Slow`**" if the number **<=** 30
  - Prints "**`Fast`**" if the number **>** 30



30 → Slow

60 → Fast

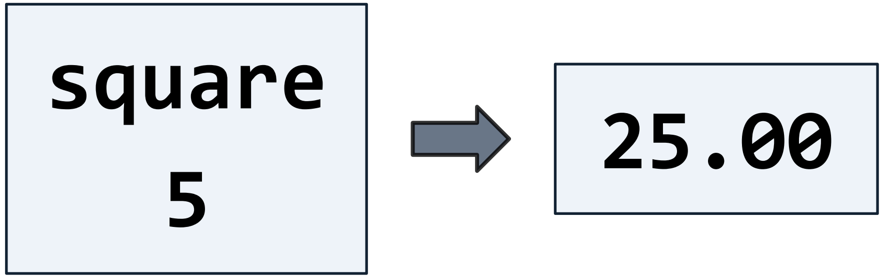
Solution: Speed Info

```
function speedInfo(speed) {
   if (speed <= 30) {
      console.log("Slow");
   } else {
      console.log("Fast");
   }
}
```

```
speedInfo(20);
speedInfo(50);
```

Problem: Area of Figures

- Write a function to **calculate figure area**, which:
  - Receives the **type** of the figure (**`string`**)
  - Receives the **size** of the figure (**`number`**)
  - Checks if the figure is **square** or **circle**
  - Prints the calculated area **formatted** to the **second decimal**



```
square
5
```

```
25.00
```

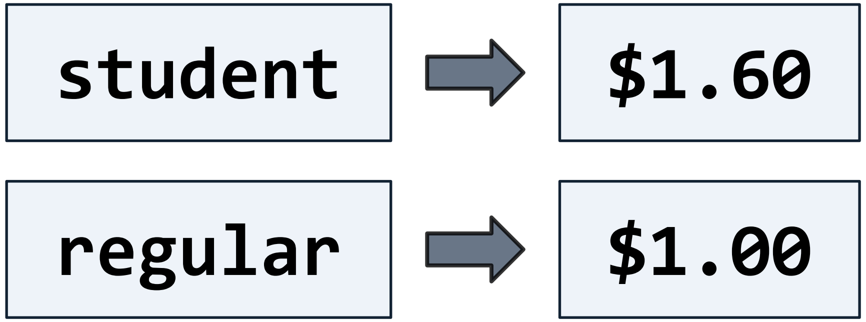
Solution: Area of Figures

```
function areaOfFigures(figure, side) {
   let area = 0;
   if (figure === 'square') {
      area = side * side;
   }
   // TODO: Add the else condition
   console.log(`${area.toFixed(2)}`);
}
```

```
areaOfFigures('circle', 5);
```

Problem: Ticket Price

- Write a function to **calculate ticket price**, which:
  - Receives a **ticket type**: either "**`student`**" or "**`regular`**"
  - Prints the **price** in the following format "**`${price}`**":
    - Student ticket price: 1.00
    - Regular ticket price: 1.60
    - For invalid type "**`Invalid ticket type!`**"



```
student → $1.60
regular → $1.00
```

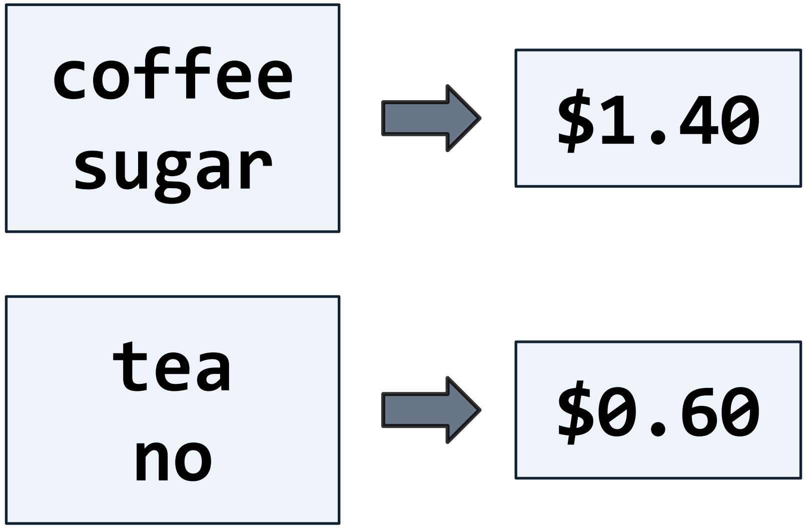
Solution: Ticket Price

```
function ticketPrice(ticketType) {
   if (ticketType === 'student') {
      console.log('$1.00');
   } else if (ticketType === 'regular') {
      console.log('$1.60');
   } else {
      console.log('Invalid ticket type!');
   }
}
```

```
ticketPrice('student');
```

Problem: Coffee Shop

- Write a function to **calculate the price for a drink**, which:
  - Receives a **drink** name: either "**`coffee`**" or "**`tea`**"
  - Receives an **extra**: either "**`sugar`**" or "**`no`**"
  - Prints the price in format "**`Final price: ${price}`**"
- Prices:
  - Coffee: 1.00
  - Tea: 0.60
  - Sugar: 0.40



```
coffee
sugar
```
→
```
$1.40
```

```
tea
no
```
→
```
$0.60
```

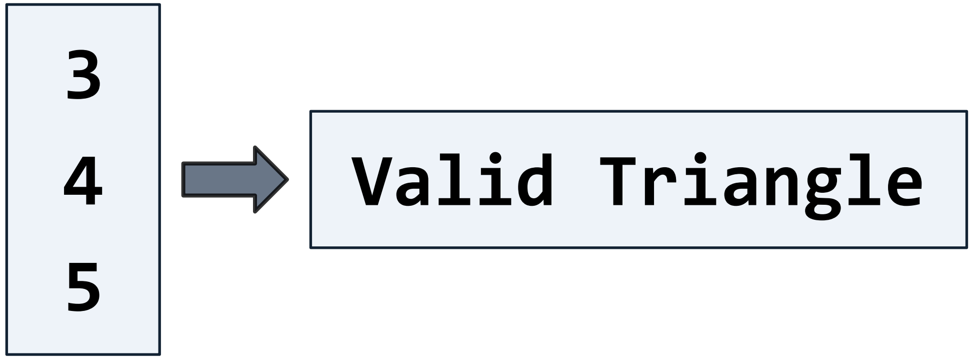
Solution: Coffee Shop

```
function coffeeShop(drink, extra) {
  let price = 0;
  if (drink === 'coffee')
    // Set the price to 1.00
  else if (drink === 'tea')
   // Set the to 0.60
  if (extra === 'sugar')
   // Increase the price by 0.40
  console.log(
   `Final price: $${price.toFixed(2)}`);
}
```

```
coffeeShop('tea', 'no');
```

Problem: Valid Triangle

- Write a function to **check is a triangle is valid**, which:
  - Receives **3 numbers**: the **sides** of a **triangle**
  - Checks if each **side** is **less** than the **sum** of the **other 2**
  - Prints "**`Valid Triangle`**" if the above condition is met
  - Prints "**`Invalid Triangle`**" otherwise



```
3
4
5
```

```
Valid Triangle
```

Solution: Valid Triangle

```
function validTriangle(a, b, c) {
  let isValidTriangle = true;
  if (a + b <= c)
    isValidTriangle = false;
  else if (a + c <= b)
    // Set isValidTriangle to false
  else if (b + c <= a)
    // Set isValidTriangle to false
  // Print the result on the console
}
```

```
validTriangle(3, 4, 5);
// Valid Triangle
```

```
validTriangle(2, 2, 5);
// Invalid Triangle
```

Summary

- Logical (Boolean) expressions
  - Comparison operators: `<`, `>`, `==`, …
- Conditional statements:
  - `if`, `if-else` and `if-else-if-…`
  - Series of `if-else` conditions
- Debugger: trace the program

Questions?

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © Kingsland University – https://kingslanduniversity.com

KINGSLAND UNIVERSITY

THANK YOU