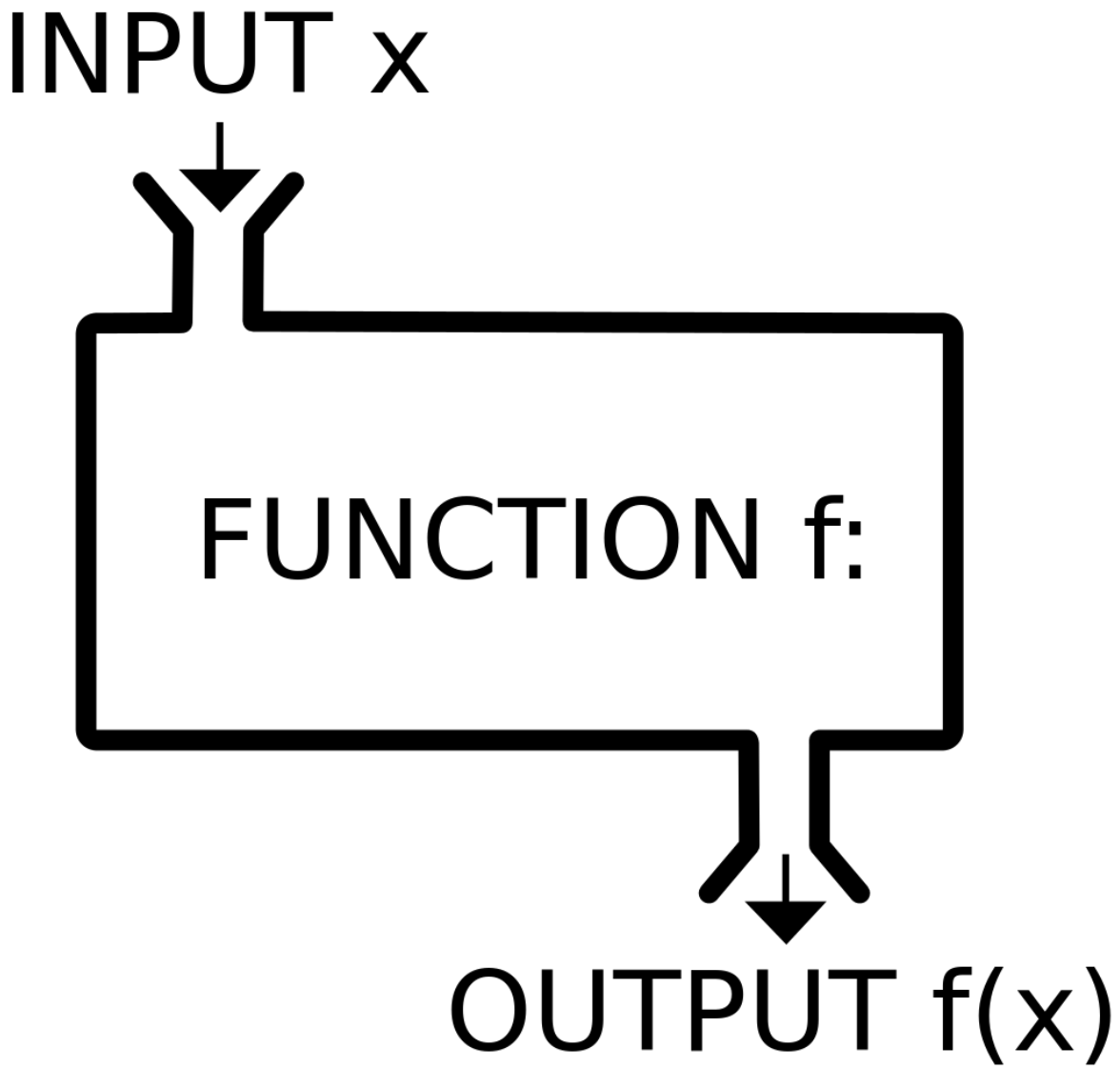
S1 Z2

Functions


INPUT x
FUNCTION f:
OUTPUT f(x)

Functions

- Basics
- Parameters
- Return

Functions

```
function sayHello() {

}
```

Functions

```
function sayHello() {
    console.log('Hello there’);
}
```

Functions

```
function sayHello() {
   console.log('Hello there');
}
sayHello();
```

Functions

```
function showValue(x){
   console.log('Value is: '+x);
}

showValue(2);
showValue('Karol');
```

Functions

```
function showSum(x,y){
    let sum = x + y;
    console.log('Sum equels :' + sum);
    console.log('Is of type :'+typeof(sum));
}

showSum(2,3);
showSum("karol",2);
showSum(2,"karol");
showSum("karol","rogowski");
```

Functions

```
let var1 = 2;
let var2 = 3;

function showSum2(x,y){
    let sum = x + y;
    console.log('Sum equels :' + sum);
    console.log('Is of type :'+typeof(sum));
    y = y+x;
    console.log(y);
}

showSum2(var1, var2);
console.log(var2);
```

Functions

```
function getSum(x,y){
    let result = x + y;
    return result;
}

let var1 = getSum(2,3);
console.log('Sum equels :' + var1);
console.log('Is of type :'+typeof(var1));

let var2 = getSum(2,'Karol');
console.log('Sum equels :' + var2);
console.log('Is of type :'+typeof(var2));

let var3 = getSum('Karol','Rogowski');
console.log('Sum equels :' + var3);
console.log('Is of type :'+typeof(var3));
```

Functions

```
function exampleFunction(){
    console.log("exampleFunction executed");
    let x = 10;
}

exampleFunction();
console.log(x);
```

Scope

- **Scope -** refers to the visibility of **variables**. In other words, which parts of your program can see or use it. Normally, every **variable** has a global **scope**. Once defined, every part of your program can access a **variable**. It is very useful to be able to limit a **variable's scope** to a single function.

Functions

```
let x = 5;
function exampleFunction() {
    console.log("exampleFunction executed");
    console.log(x);
}

exampleFunction();
console.log(x);
```

Functions

```
let x =5;

function exampleFunction(){
    console.log("exampleFunction executed");
    let x = 10;
    console.log(x);
}

exampleFunction();
console.log(x);
```

Functions

```
let x =5;

function exampleFunction(){
    console.log("exampleFunction executed");
    x = 10;
    console.log(x);
}

exampleFunction();
console.log(x);
```

Functions

```
let x =5;

function exampleFunction(){
    let x =1;
    console.log("exampleFunction executed");
    x = 10;
    console.log(x);
}

exampleFunction();
console.log(x);
```


Objects

Objects

- Basics
- Objects + Functions
- Grouped Objects
- Out of the box

Objects

```
let book = {
   title: 'LOTR',
   pages: 2745,
   hardcover: true
}
```

Objects

```
let book = {
   title: 'LOTR',
   pages: 2745,
   hardCover: true
};

console.log(book.title);
console.log(book.pages);
console.log(book.hardCover);
```

Objects

```
let book = {
    title: 'LOTR',
    pages: 2745,
    hardCover: true
};

function showBookInfo(bookObject){
    console.log(bookObject.title);
    console.log(bookObject.pages);
    console.log(bookObject.hardCover);
}

showBookInfo(book);
```

Objects

```
let book = {
    title: 'LOTR',
    pages: 2745,
    hardCover: true
};

function changeCover(book){
    book.hardCover = !book.hardCover;
    console.log('Cover changed');
}

changeCover(book);
showBookInfo(book);
```

Objects

```
let books = [
    {
        title: 'LOTR',
        pages: 2745,
        hardCover: true
    },
    {
        title: 'Witcher',
        pages: 1266,
        hardCover: false
    },
    {
        title: 'Sherlock Holmes',
        pages: 1950,
        hardCover: true
    }
];
```

Objects

```
for(let i = 0; i < books.length; i++){
    showBookInfo(books[i]);
}

books.forEach(function(book) {
    showBookInfo(book);
});
```

Out of the box

Math

Date

String

Number

Error

Function

Math

```
Math.max();
Math.min();
Math.floor();
Math.pow();
Math.PI;
Math.SQRT2;
Math.LOG10E;
Math.E;
```

Date

```
let dateObj = new Date();
console.log(dateObj);
console.log(dateObj.toDateString());
console.log(dateObj.toLocaleDateString());
dateObj.setMonth(11);
console.log(dateObj);
dateObj.setMonth(12);
console.log(dateObj);
```

String

```
let str = "Karol";
console.log(str.substr(2, 2));
console.log(str.repeat(3));
console.log(str.includes("o"));
console.log(str.anchor("test"));
console.log(str.length);
```

Number

```
let k = 2.12345;
console.log(k.toExponential(3));
console.log(k.toFixed(3));
console.log(k.toPrecision(3));

k = 4;
console.log(Number.isInteger(k));
console.log(Number.isNaN(k));
console.log(Number.isSafeInteger(k));
```