Functions

What's a function's function?

WEB PROGRAMMING FUNDAMENTALS
web design and development
full sail university

The 4 Types

- Sequential
- Conditional
- Repetitive
- Reusable

The 4 Types

- Sequential - Expressions
- Conditional
- Reusable
- Repetitive

The 4 Types

- Sequential - Expressions
- Conditional - If statements, ternaries
- Reusable
- Repetitive

The 4 Types

- Sequential - Expressions
- Conditional - If statements, ternaries
- Repetitive - Loops
- Reusable

The 4 Types

- Sequential - Expressions
- Conditional - If statements, ternaries
- Repetitive - Loops
- Reusable

Functions

- Why do we need them?
  - DRY - Don't Repeat Yourself!
  - They allow us to have code that is written once, but able to be used again and again by the machine.
  - Organizes Code

Functions

- Functions store stuff
  - Variables - store values
  - Functions - store blocks of code

Like a Factory

- Ways to think about functions:
  - Like modular factories.
  - Stuff can go in, stuff can come out.

Like a Recipe

- Ways to think about functions:
  - References to another recipe you might use many times.
  - Instead of printing the same recipe in many pages of a recipe book, why not put a reference to the recipe. (Ex: Check out the frosting recipe on page 250)

Reusing Code

- In Games:
  - Functions allow you to have the code in one place and run that code as many times as you want.
  - Code for shooting - used again and again

A Basic Function

Structure and Use

WEB PROGRAMMING FUNDAMENTALS
web design and development
full sail university

Basic Structure

```
function functionName(){
  //code the function runs
}
```

Basic Structure

```
function functionName(){
   //code the function runs
}
```

- Functions have names so we can refer to them later.

Basic Structure

```
function functionName(){
   //code the function runs
}
```

- Parenthesis are for parameters

Basic Structure

```
function functionName(){
   //code the function runs
}
```

- Curly braces are for blocks of code

Watch Out

- MUST have the parenthesis if you want the function to be executed.
  - WRONG: var myVar = myFunction;
  - RIGHT: var myVar= myFunction();
- Without the parenthesis, you are assigning a reference to that function, as opposed to a reference to its result. As in:
  - var myNewFn= myFunction;
    myNewFn();