JavaScript

Introduction to functions

As I was going to St. Ives, I met a man with seven wives.
Every wife had seven sacks, and every sack had seven cats, and every cat had seven kittens.
Kittens, cats, sacks, and wives, how many were going to St. Ives?

Your task: write a program in JavaScript to find the answer!

Solution

1. 7 + (7 * 7) + (7 * 7 * 7) + (7 * 7 * 7 * 7)
2. Make code more expressive > help readers understand it
    - *How can we do that?*
3. **Give names to its parts**:
    - We have 4 numbers; each means something particular. Use the console:

```
>   wives = 7
>   sacks = 7 * 7
>   cats = 7 * 7 * 7
>   kittens = 7 * 7 * 7 * 7
>   total = wives + sacks + cats + kittens
>   2800
```

Solution

4. ...but those " *7 " also mean something!
   - There is a relationship between the number of wives
     and the number of sacks, the number of cats, etc...

5. Our statements don't say anything about that.
   - Let's fix it!

```
>   wives = 7
>   sacks = 7 * wives
>   cats = 7 * sacks
>   kittens = 7 * cats
>   total = wives + sacks + cats + kittens
>   2800
```
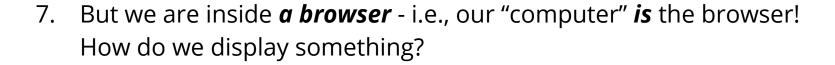
Solution

4. Now let's turn this code into a program!
   - Download ives.html
   - Create a file ives.js and link it to ives.html
   - Place your code into ives.js + **make it valid JavaScript**

```javascript
var wives = 7;

var sacks = 7 * wives;

var cats = 7 * sacks;

var kittens = 7 * cats;

var total = wives + sacks + cats + kittens;
```

5. But how do we "see" the answer?
   - Console: a "conversation" with an interpreter: read > evaluate > print > do it again
   - Program: set of explicit instructions. To display anything, we have to say that explicitly!

Solution

6. In a traditional programming setting, it's easy:
   - in python: print(total)
   - in Windows (or Mac): click the executable file/icon

7. But we are inside **a browser** - i.e., our "computer" **is** the browser! How do we display something?

8. We use our web page!
   - we use alert(total);
   - or something better >>

...A reminder

- We can use JavaScript to access content:
    - document.querySelector("**your-selector**");
    - document.querySelector("h1"); returns the first H1 element
    - myHeader = document.querySelector("h1");

        assigns the first H1 element to variable **myHeader**

- We can use JavaScript to modify content:
    - document.querySelector("h1").innerText = "whatever text you like";
    - sets the text of the first H1 element to this text: whatever you like

    - document.querySelector("h1").innerHTML = "whatever HTML you like";
    - sets the inner HTML of the first H1 element to: whatever **HTML** you like

Solution


```javascript
var wives = 7;
var sacks = 7 * wives;
var cats = 7 * sacks;
var kittens = 7 * cats;
var total =  wives + sacks + cats + kittens;


//get the paragraph where you want to display the result
var target = document.querySelector(".answer");


//get the current text of that paragraph
var currentText = target.innerText;


//add a space and the result to the current text and assign to the paragraph
target.innerText = currentText + " " + total;
```

A new twist…

- ## What if there are more wives?

```
var wives = 7;
var sacks = 7 * wives;
var cats = 7 * sacks;
var kittens = 7 * cats;
var total1 =  wives + sacks + cats + kittens;
var target1 = document.querySelector(".answer1");
var currentText = target1.innerText;
target1.innerText = currentText + " " + total1;


var wives = 8;
var sacks = 8 * wives;
var cats = 8 * sacks;
var kittens = 8 * cats;
var total2 =  wives + sacks + cats + kittens;
var target2 = document.querySelector(".answer2");
var currentText = target2.innerText;
target2.innerText = currentText + " " + total2;
```

```
var wives = 9;
var sacks = 9 * wives;
var cats = 9 * sacks;
var kittens = 9 * cats;
var total3 =  wives + sacks + cats + kittens;
var target3 = document.querySelector(".answer3");
var currentText = target3.innerText;
target3.innerText = currentText + " " + total3;
```

...as per your textbook:

This is turrible!

Let's use a function!

- A function:
  - is a named piece of code
  - that groups statements together
  - and makes code reusable

- How to use a function:
  1. Declare the function
  2. Call the function

Simple example of a function

//declare the function
```
function saySomething() {
    alert("something");
}
```

//call the function
```
saySomething();
```

An example of a function that takes arguments

//declare the function

```
function saySomething(stuffToSay) {
    alert(stuffToSay);
}
```

//call the function

```
saySomething("I have a lot to say!");
```

An example of a function that returns data

//declare the function

```
function saySomethingTwice(stuffToSay) {
    return stuffToSay + " " + stuffToSay;
}
```

//call the function

```
var result = saySomethingTwice("I have a lot to say!");
```

```
//now do anything you want with the result!
```

A Better solution to our problem


```
function countEverything() {
    var wives = 7;
    var sacks = 7 * wives;
    var cats = 7 * sacks;
    var kittens = 7 * cats;
    var total =  wives + sacks + cats + kittens;
    return total;
}

var target = document.querySelector(".answer");
var currentText = target.innerText;
target.innerText = currentText + " " + countEverything();
```

A MUCH better solution to our problem


```javascript
function countEverything(times) {
    var wives = times;
    var sacks = times * wives;
    var cats = times * sacks;
    var kittens = times * cats;
    var total =  wives + sacks + cats + kittens;
    return total;
}
var target1 = document.querySelector(".answer1");
var currentText = target1.innerText;
target1.innerText = currentText + " " + countEverything(7);

var target2 = document.querySelector(".answer2");
var currentText = target2.innerText;
target2.innerText = currentText + " " + countEverything(42);
```