

G9 TEJ10 Exploring Computer Technology

Introduction to Programming with JavaScript

The following examples and problem sets are intended as a lean introduction to programming using the language that drives the internet, namely, JavaScript.

Requirements

The tools needed are

1. **Brackets**, as a text editor. You can download it from here: <http://brackets.io/>
2. **Our HTML Live Editor** which you can download from here: <http://msantos.sdf.org/htmlLiveEditor.html>

Workflow

Use the HTML-Live editor for easy practising and solving the problems.

When you finish a program or solve a problem, **save the code**:

1. Select and Copy all your code you wrote in the HTML-Live editor.
2. Open Brackets.
3. Click on the Menu **File** and choose the option **New File**.
4. Click on the Menu **File** and choose the option **Save File as**.
5. Give it a name that is meaningful. Example: If the topic is **while-loops** the name could be **while-loop.html**
6. Paste the code you copied
7. Click on the Menu **File** and choose the option **Save File**. Alternatively, press the keyboard shortcut **Ctrl+S**.

Comments

1. **Inline comment**: Anything *following* a double forward slash `//` and until the end of the line will be ignored by the computer.
2. **Multiple-lines comment**: Anything between `/*` and `*/` will be ignored by the computer.

Arithmetics with JavaScript

1. Additions and substractions

```
/*  
    This is a multi-line comment.  
  
    The result of each print statement is given as a comment  
    following that statement  
*/  
print( 3 + 5 ) //This prints 8  
  
print( 18 - 20 ) //This prints -2
```

2. Multiplication and division

```
print( 3 * 5 ) //This prints 15  
  
print( 18 / 20 ) //This prints 0.9
```

3. Square root and powers

```
print( Math.sqrt(16) ) //This prints the square root of 16, which is 4  
  
print( Maht.pow(2,3) ) //This prints 8  
  
print( Maht.pow(2,4) ) //This prints 16  
  
print( Maht.pow(3,2) ) //This prints 9  
  
print( Maht.pow(3,3) ) //This prints 27  
  
print( Maht.pow(3,4) ) //This prints 81  
  
print( Maht.pow(3,0) ) //This prints 1  
  
print( Maht.pow(2,0) ) //This prints 1  
  
print( Maht.pow(137,0) ) //This prints 1
```

Assignments: Variables and Values

When programming it is essential to be able to **store** data in the memory of the computer for later use.

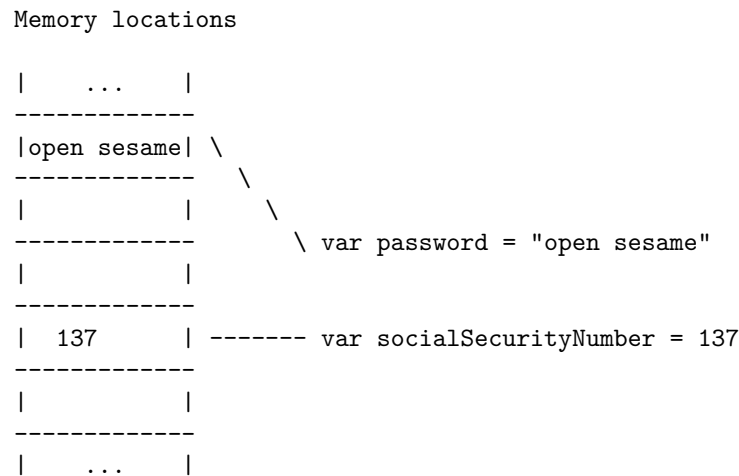
In order to store data in the memory of a computer we use **variables**.

Variables are just *labels* that point to some location in the memory of the computer where some data is stored.

We define a variable using the keyword **var** followed by the label we want to use, then an equal sign = and then the actual value that we want to store in the memory.

Giving a value to a variable is called **assigning a value** to that variable.

Having a label for some data in memory allows us to retrieve that data anytime we want by just using the label. When the computer sees the label, it will grab the data stored in the memory location where that label points to.



The label **password** points to a memory location that stores the text ‘open sesame’.

The label **socialSecurityNumber** points to a memory location that stores the value 137.

Numbers

1. Define a variable labeled **x** and assign it the value 137

Solution:

```
var x = 137
```

2. Define a variable labeled **x** and assign it the value 137. Define a variable **y** and assign it the value 10. Finally, define a variable called **z** and assign it the formula $x * y - y$

Solution:

```
var x = 137
var y = 10
var z = x*y - y
```

Strings

1. Assign the string value of `John` to a variable called `name` and the number value of `2018` to another called `year`. Print a message using these two variables saying `I, John, live in the year 2018`. Make sure that you include the right amount of spaces and all punctuation marks.

Solution:

```
var name = "John"
var year = 2018
print( "I, "+ me +", live in the year "+x)
```

Input/Ouput: Prompting the user and displaying information

1. Write a program that prompts the user for her/his name. Assign it to a variable called `name`. Define another variable called `year` and assign it the value `2018`. Imagine the user's name is 'John'. Then your program should print a message using these two variables saying `I, John, live in the year 2018`. Make sure that you include the right amount of spaces and all punctuation marks.

Solution:

```
var me = prompt("Please, enter your name: ")
var x = 2018
print( "I, "+ me +", live in the year "+x)
```

2. Write a program that asks the user for two numbers. Store them in two variables called `x` and `y`. Then print a message with their sum, in the following way: Say, 'x=100' and 'y=37', then your program should print '100+37=137'

Solution:

```
var x = Number(prompt("Enter a number: "))
var y = Number(prompt("Enter a second number: "))

print(x+" "+y+"="+ (x+y))
```

3. **Cashier Machine:** Write a program that asks the user for three items and their costs. Then prints each item followed by its cost. Then prints a

short separation line like '——'. Then prints the total cost, which is the sum of all three costs, and the amount of taxes, which are 13% of the total cost. Imagine the total is '137'. Then it should print 'Total: 137' and the following line should be 'Tax: 17.81'. Finally, your program should print another separation line followed by the bill amount. This is the sum of the total amount plus the taxes. Following the example it would print 'Bill: 153.61'.

Solution:

```
var item1_name = prompt("Name of Item 1? ")
var item1_cost = Number( prompt("Cost of "+item1_name+"?") )

var item2_name = prompt("Name of Item 2? ")
var item2_cost = Number( prompt("Cost of "+item2_name+"?") )

var item3_name = prompt("Name of Item 3? ")
var item3_cost = Number( prompt("Cost of "+item3_name+"?") )

print(item1_name+" "+item1_cost)
print(item2_name+" "+item2_cost)
print(item3_name+" "+item3_cost)

print("-----")

var total = item1_cost + item2_cost + item3_cost

var tax = total * 13 / 100

var bill = total + tax

print("Total: "+total)
print("Tax: "+tax)

print("-----")

print("Bill: "+bill)
```

If-Then-Else Conditionals

1. Guess what will be printed in each case. Assume `x` is 70 and `name` is 'johny'.

```
var x = 70

if ( x < 10 ) {
```

```

    print(x+" is less than 10") //Nothing
}

if ( x >= 3 ) {
    print(x+" is larger or equal than 3") // '70 is larger or equal than 3'
}

if ( x <= 3 ) {
    print(x+" is less or equal than 3") //Nothing
}

if ( x == 5) {
    print(x+" is equal to 5") //Nothing
}

if ( ! (x > 5) ) {
    print(x+" is NOT larger than 5") //Nothing
}

if ( x != 5) {
    print(x+" is NOT equal to 5") // '70 is NOT equal to 5'
}

if ( ! (x == 5) ) {
    print(x+" is NOT equal to 5") // '70 is NOT equal to 5'
}

if ( x < 10 && x > 5) {
    print(x+" is less than 10 AND larger than 5") //Nothing
}

if ( x < 10 || x > 5) {
    print(x+" is less than 10 OR larger than 5") // '70 is less than 10 OR larger than 5'
}

var name = "johny"

if ( name != "john") {
    print("Hiya, "+name) // 'Hiya, johny'
}

if ( name == "john") {
    print("what's up, yo ")
} else{
    print("Good morning, Mr. "+name) // 'Good morning, Mr. johny'
}

```

```

if ( name == "john") {
    print("what's up, yo ")
}
else if (name == "johnny" ) {
    print("Good morning, Mr. "+name)
}
else {
    print("Your name is neither John nor Johnny") // 'Good morning, Mr. johnny'
}

```

2. **Game: Guess it! I.** Write a program that prompts the user to guess a hidden number. If the user's answer is wrong, show a dialog window saying "Wrong!". If it is correct, show a dialog windows saying "Correct!".

Solution:

```

var hidden = 3

var userAnswer = Number( prompt("Guess a number: ") )

if ( userAnswer == hidden ) {
    alert("Correct!")
}
else{
    alert("Wrong!")
}

```

2. **Game: Guess it! II.** Write a program that prompts the user to guess a hidden number. This time, the user has 3 chances to guess the number correctly. Say the hidden number is 12. If the user's guess is 20, show a dialog window saying "It's smaller!"; if the user's guess is 8, show a dialog window saying "It's larger!".

If the user guesses it wrong all 3 times then show a dialog window saying "Game Over!".

If at any time the user guesses the correct value, show a dialog window saying "Correct!" and the program should stop.

Solution:

```

var guess = "Guess a number"
var correct = "Correct!"
var isSmaller = "It's smaller!"
var isLarger = "It's larger!"
var gameOver = "Game Over!"

var hidden = 33

```

```

var userAnswer = Number( prompt("Guess a number: ") )

if ( userAnswer == hidden ) {
    alert(correct)
}
else{
    if ( userAnswer < hidden ) {
        alert( isLarger )
    }
    else{
        alert( isSmaller )
    }
    userAnswer = Number( prompt("Guess a number: ") )
    if ( userAnswer == hidden ) {
        alert(correct)
    }
    else{
        if ( userAnswer < hidden ) {
            alert( isLarger )
        }
        else{
            alert( isSmaller )
        }
        userAnswer = Number( prompt("Guess a number: ") )
        if ( userAnswer == hidden ) {
            alert( correct )
        }
        else{
            alert( gameOver )
        }
    }
}
}

```

While-loops

1. Write a while-loop that counts from 0 to 9 and each time prints “Count =” and then the value of the current count. Example:

```

Count = 0
Count = 1
Count = 2
...

```

Solution:

```

var count = 0

```



```

while ( count < 10 ) {
    print("Count =" + count )
    count = count + 1          //This is the loop, update rule!
}

```

2. Write a while-loop that prints the happy-birthday song so that we repeat statements the least number of times possible.

Solution:

```

var i = 0

while ( i <= 2 ) {
    print( "Happy birthday to you" )
    i = i + 1
}

print( "Happy birthday dear Julia" )
print( "Happy birthday to you" )

```

3. **Game: Guess it! III.** We want to rewrite our game *Guess it!* using while-loop. It makes sense. After all the game consists in repeating at most 3 times the same question (prompting the user to guess a number) and possible actions that need to be taken are always the same. Would it be possible to tell the computer to *do all that 3 times at most*? Well, there is, namely using while-loops.

Problem: Rewrite the game *Guess it!* so that the code contains only 1 single line prompting the user for a number, and yet it gives the user 3 chances at most to guess the hidden number.

Solution:

```

var secretNumber = 3

var counter = 1

while ( counter <= 3 ) {
    var userAnswer = Number( prompt( "Guess a number: " ) )

    if ( counter == 3 && userAnswer != secretNumber ) {
        alert( "Game over!" )
        break          // 'break' will force stop the while-loop altogether
    }

    if ( userAnswer == secretNumber ) {
        alert( "Correct!" )
        break          // User guessed won. Let's stop the while-loop here.
    }
}

```

```

else if ( userAnswer > secretNumber ) {
    alert("It's smaller. Try again")
}
else {
    alert("It's larger. Try again")
}

counter = counter + 1 // Never forget this! Update the loop counter!!
}

```

4. Write a while-loop to calculate the sum $1 + 2 + 3 + 4 + 5 + \dots + 100$. Make your program print the final result.

Solution:

```

var sum = 0

var i = 1
while ( i <= 100 ){
    sum = sum + i
    i = i + 1
}
print( sum )

```

5. Write a while-loop to calculate the sum $2 + 4 + 6 + 8 + 10 + \dots + 200$. Make your program print the final result.

Solution:

- First we need to find the pattern:** The numbers we need to add are all the even numbers up to 200. We need to find the **pattern** relating the number position and its value. Let's do it by filling in the following table

i	number	Pattern
1	2	?
2	4	?
3	6	?
4	8	?
5	10	?
6	12	?

The **pattern** is the answer to the following question: **How can I get each number of column 2 from the corresponding value of column 1?** For instance, how can we get 4 from 2? One possible answer could be *by taking the square of 2*, namely, $2^2 = 4$. However, *this rule would not apply for the other rows*: for the third row it would mean $3^2 = 9$ which does not

give us 6! Whence, *taking the square is not the right pattern.*

The **right pattern is multiplying by 2**. We write this as a formula with the counter variable i : $2 * i$

i	number	Pattern
1	2	$2*i$
2	4	$2*i$
3	6	$2*i$
4	8	$2*i$
5	10	$2*i$
6	12	$2*i$

2. **Then we need to find the limit value for the counter i :** The limit value for i is that one that makes the pattern equal to the last term.

$$2 * i = 200$$

The solution is $i = 100$. This is the limit value.

3. **We use the pattern for *updating the sum*:** Now we can write our program. In the line `sum = sum + ...` we will use the expression for the pattern we just found, that is, it will be `sum = sum + 2*i`:

```
var sum = 0

var i = 1
while ( i <= 100 ){
    sum = sum + 2*i
    i = i + 1
}
print( sum )
```

6. Write a while-loop to calculate the sum $1 + 3 + 5 + 7 + 9 + \dots + 199$. Make your program print the final result.

Solution:

1. **First we need to find the pattern:** The numbers we need to add are all the odd numbers up to 199. We need to find the **pattern** relating the number position and its value. Let's do it by filling in the following table

i	number	Pattern
1	1	?
2	3	?
3	5	?
4	7	?

i	number	Pattern
5	9	?

How can we obtain the numbers of column 2 from those of column 1?
 What formula relates, say 5 with 3? o 7 with 4?

The pattern in this case is $2 * i - 1$

i	number	Pattern
1	1	2i-1
2	3	2i-1
3	5	2i-1
4	7	2i-1
5	9	2i-1

2. **Then we need to find the limit value of i:** The limit value is that value of i that makes the pattern equal to the last term

$$2 * i - 1 = 199$$

The solution is $i = 100$.

3. **We use the pattern for *updating* the sum:** Now we can write our program. In the line `sum = sum + ...` we will use the expression for the pattern we just found, that is, it will be `sum = sum + 2*i-1`:

```
var sum = 0

var i = 1
while ( i <= 100 ){
    sum = sum + 2*i-1
    i = i + 1
}
print( sum )
```

7. Write a program that calculates the sum $2+6+12+20+30+42+56+\dots+110$ and prints the final result.

Solution

The pattern in this case is $i * (i + 1)$

i	number	Pattern
1	2	$i * (i + 1) = 1 * (1 + 1) = 2$
2	3	$i * (i + 1) = 2 * (2 + 1) = 6$
3	5	$i * (i + 1) = 3 * (3 + 1) = 12$

i	number	Pattern
4	7	$i * (i + 1) = 4 * (4 + 1) = 20$
5	9	$i * (i + 1) = 5 * (5 + 1) = 30$

The limit value of i is that which makes the pattern equal to 110, that is

$$i * (i + 1) = 110$$

We can guess the solution by trying a few values. The limit value of i is $i = 10$

Whence the program is

```
var sum = 0

var i = 1
while ( i <= 10 ){
    sum = sum + i*(i+1)
    i = i + 1
}
print( sum )
```

8. Write a program that calculates the sum $1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots + \frac{1}{1024}$ and prints the final result.

Solution: In order to find the pattern, *sometimes it is easier to make the counter start by 0!*

The pattern in this case is $\frac{1}{2^i}$

i	number	Pattern
0	1	$1/2^i = 1/2^0 = 1$
1	1/2	$1/2^i = 1/2^1 = 1/2$
2	1/4	$1/2^i = 1/2^2 = 1/4$
3	1/8	$1/2^i = 1/2^3 = 1/8$
4	1/16	$1/2^i = 1/2^4 = 1/16$

The limit value of i is that which makes the pattern equal to $1/1024$, that is

$$1/2^i = \frac{1}{1024}$$

which is the same as saying that it has to be

$$2^i = 1024$$

We can guess the solution by trying a few values. The limit value of i is $i = 10$

Whence the program is

```
var sum = 0

var i = 0
while ( i <= 10 ){
    sum = sum + 1/Math.pow(2,i)
    i = i + 1
}
print( sum )
```