Telerik Academy

Loops

Execute Blocks of Code Multiple Times

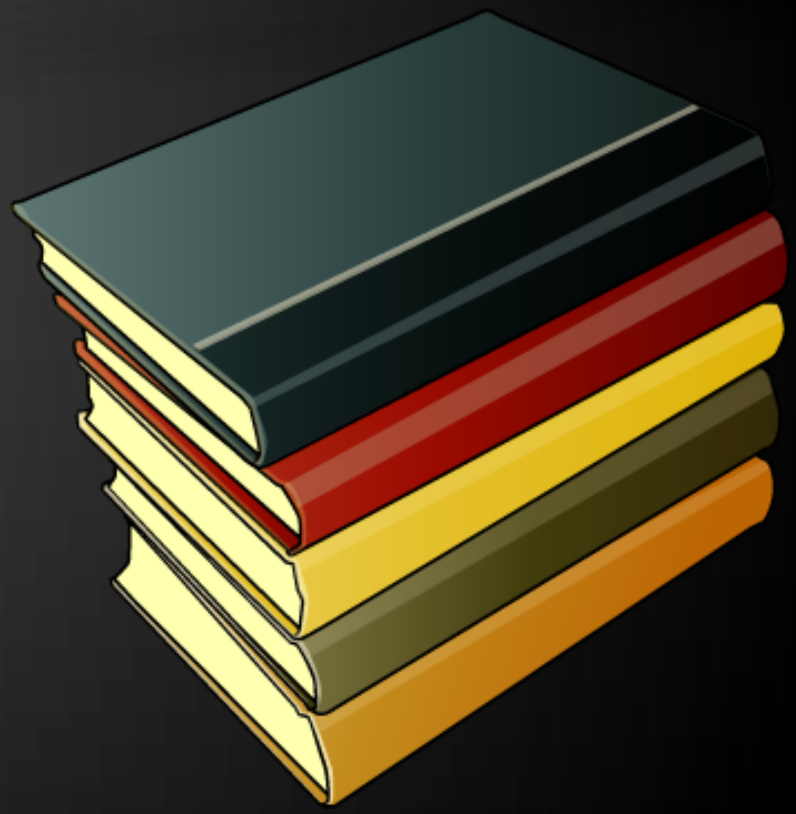
Telerik Software Academy

Learning & Development Team

http://academy.telerik.com

Telerik Academy

Table of Contents



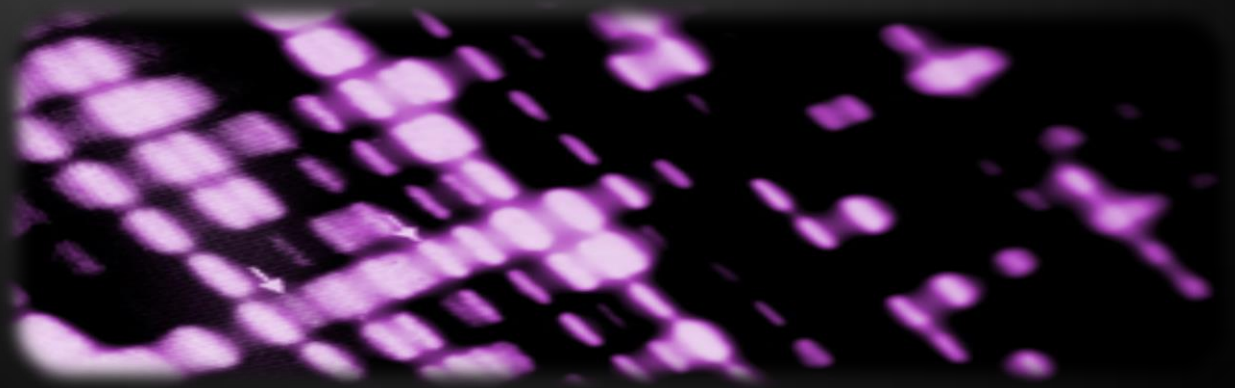
What Is Loop?

- A loop is a control statement that allows repeating execution of a block of statements
  - May execute a code block fixed number of times
  - May execute a code block while given condition holds
  - May execute a code block for each member of a collection
- Loops that never end are called an infinite loops

Telerik Academy

Using while(…) Loop

Repeating a Statement While Given Condition Holds



Telerik Academy

How To Use While Loop?

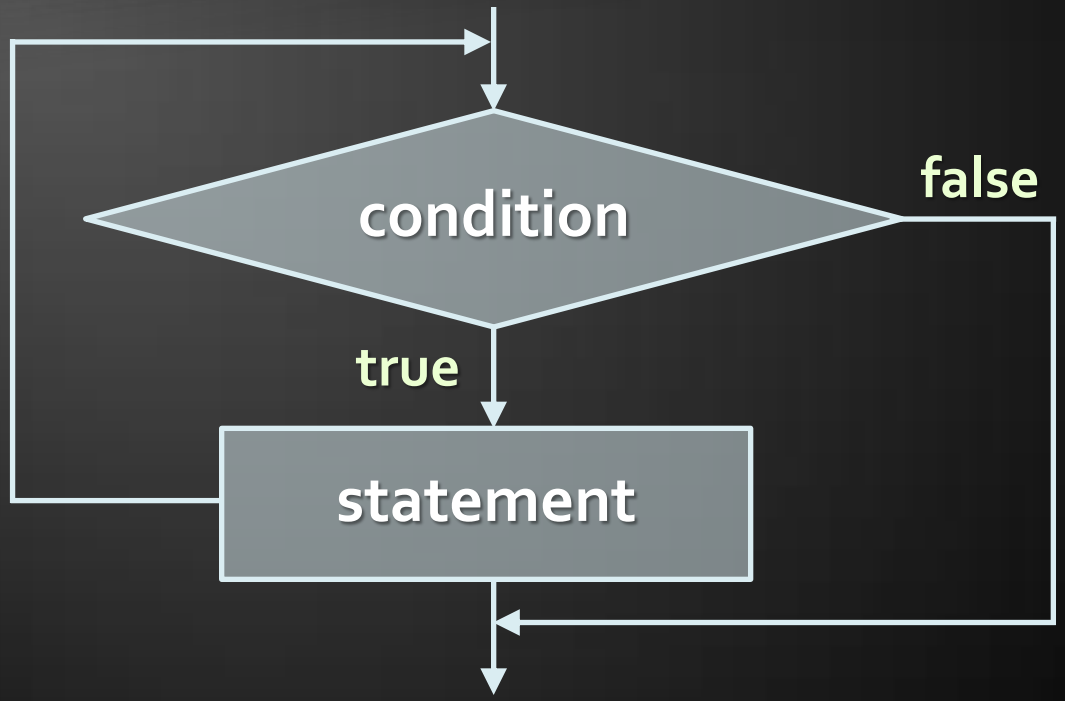
- The simplest and most frequently used loop

```
while (condition) {
    statements;
}
```

- The repeat condition
  - Also called loop condition
  - Is not necessary true or false
  - Is evaluated to `true` or `false`
    - 5, "non-empty", etc.. are evaluated as true
    - 0, "", null are evaluated as false

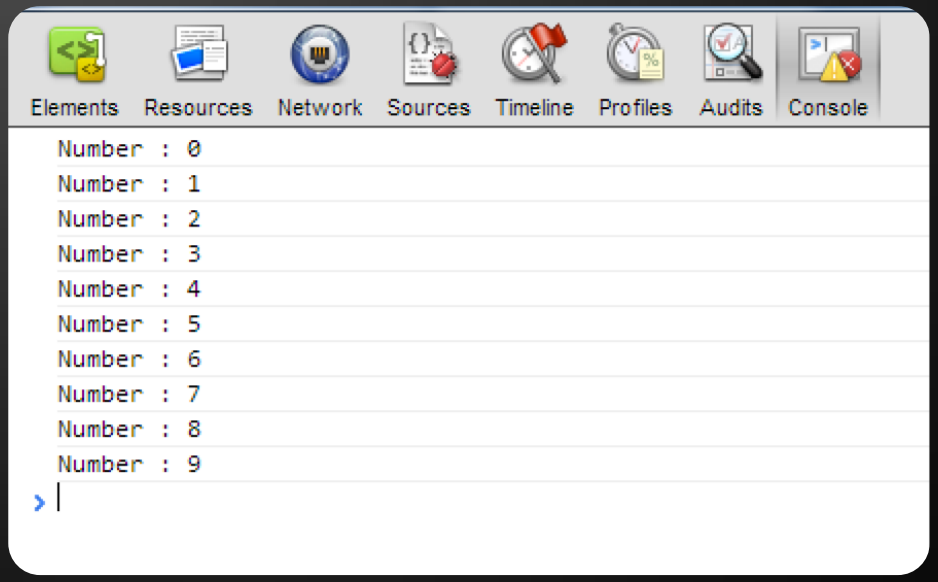
Telerik Academy

While Loop – How It Works?

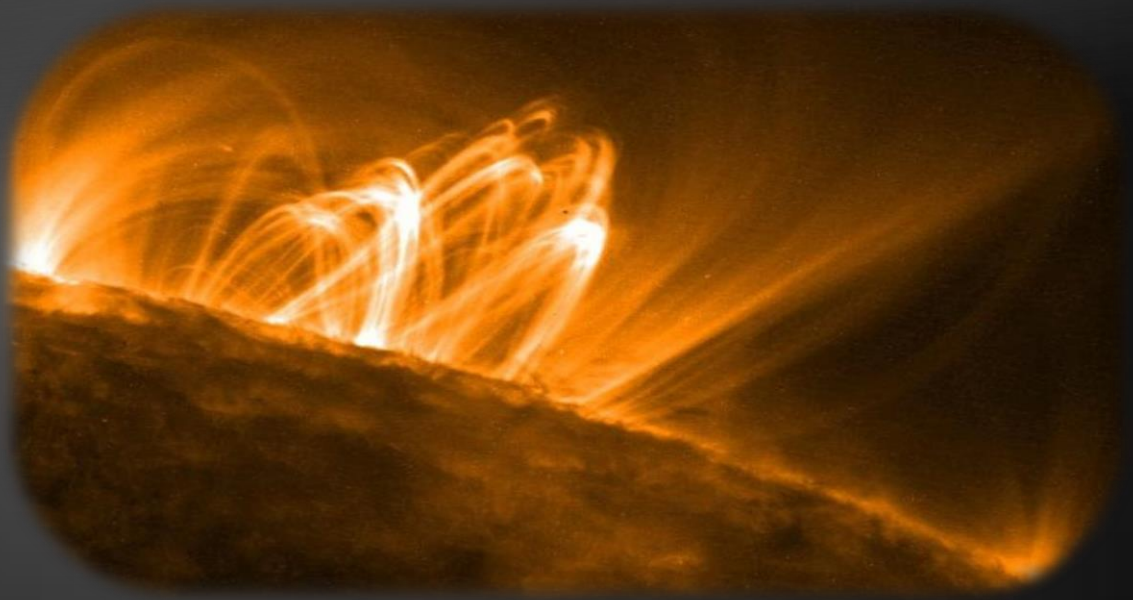


While Loop – Example

```
var counter = 0;
while (counter < 10) {
    console.log('Number : ' + counter);
    counter++;
}
```



Telerik Academy



while(…)

Examples

Sum 1..N – Example

- Calculate and print the sum of the first N natural numbers

```
var nStr = read('input-tb');
var n = parseInt(nStr);
var number = 1;
var sum = 1;
var result = 'The sum 1';
while (number < n) {
    number++;
    sum += number ;
    result += '+' + number;
}
result += ' = ' + sum;
print('console-out',result);
```

Prime Number – Example

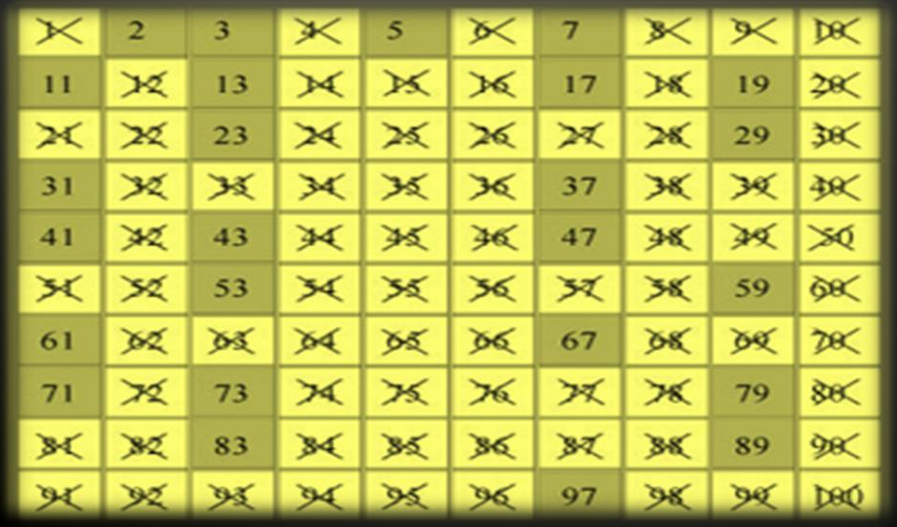
- Checking whether a number is prime or not

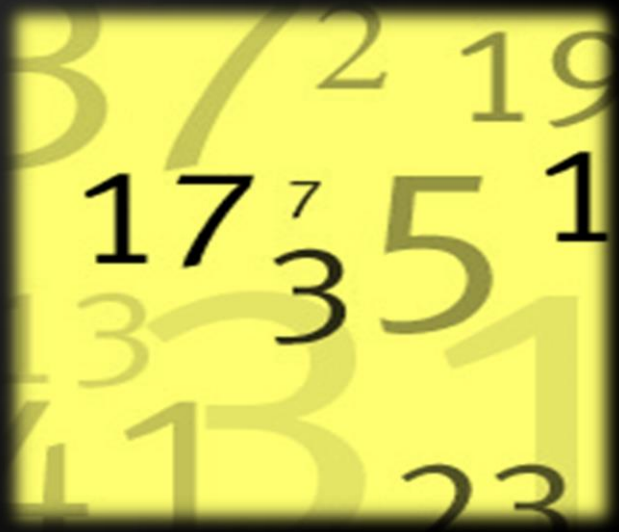
```
var numberStr = read('input-tb');
var number = parseInt(numberStr);
var divider = 2;
var maxDivider = Math.sqrt(number);
var prime = true;
while (prime && (divider <= maxDivider)) {
    if (number % divider == 0) {
        prime = false;
    }
    divider++;
}
print('output-tb',prime);
```

Telerik Academy

While Loop

Live Demo



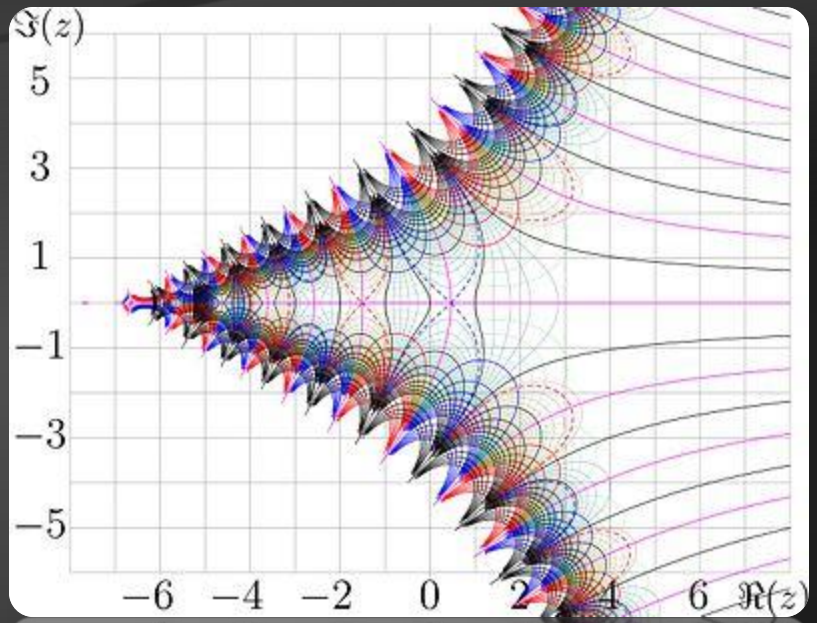


Using break Operator

- break operator exits the inner-most loop

```
var numberStr = read('input-tb');
var n = parseInt(numberStr);
var fact = 1;
var factStr = 'n! = ';
while (true) {
    if (n == 1)
        break;
    factStr += n + '*'
    fact *= n;
    n--;
}
factStr += '1 = ' + fact;
print('output-tb',factStr);
```

Telerik Academy



Calculating Factorial

Live Demo

Telerik Academy

do { … } while (…) Loop

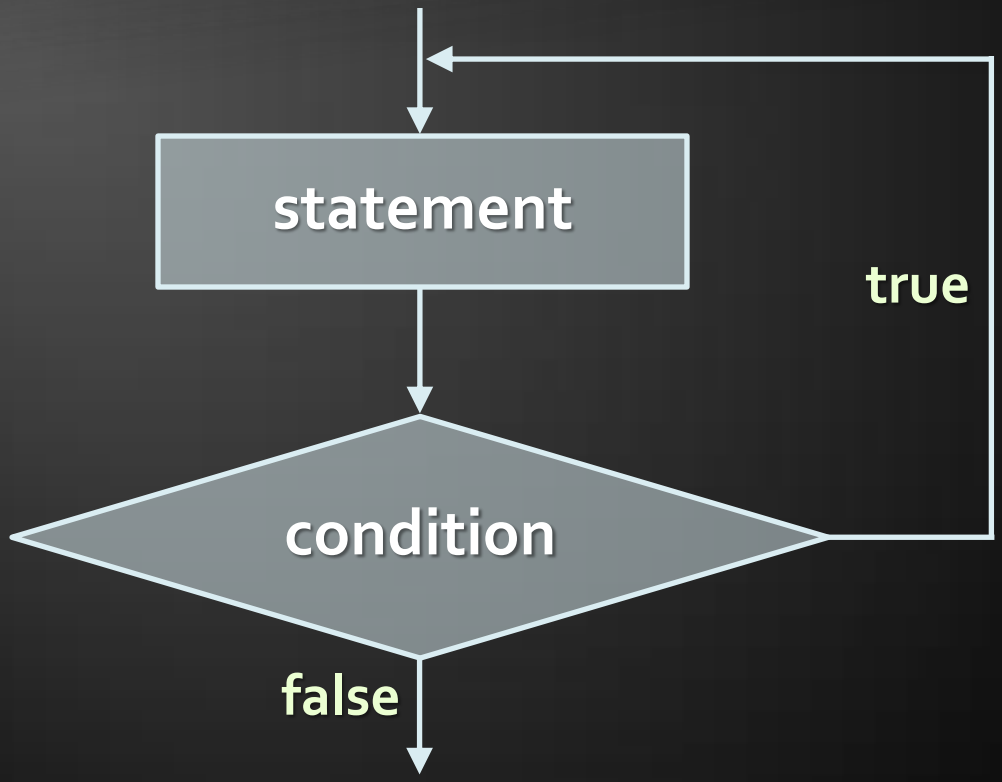
Using Do-While Loop

- Another loop structure is:


```
do {
    statements;
}
while (condition);
```

- The block of statements is repeated
  - While the boolean loop condition holds
- The loop is executed at least once

Do-While Statement



Telerik Academy

do { … } while (…)

Examples

Factorial – Example

- Calculating N factorial

```
var fact = 1;
var factStr = 'n! = ';
do {
    fact *= n;
    factStr += n + '*'
    n--;
} while (n);
factStr += ' = ' + fact;
print('output-tb',factStr);
```

Product[N..M] – Example

- Calculating the product of all numbers in the interval [n..m]:

```
var number = n;
var product = 1;
var productStr = '';
do {
    product *= number;
    productStr += number;
    if (number != m) {
            productStr += '*';
    }
    number++;
} while (number <= m);
productStr += ' = ' + product;
print('output-tb', productStr);
```

Telerik Academy

do-while Loop

Live Demo

Telerik Academy

for Loops

For Loops

- The typical for loop syntax is:

```
for (initialization; test; update) {
    statements;
}
```

- Consists of
  - Initialization statement
  - Test expression that is evaluated to boolean
  - Update statement
  - Loop body block

Telerik Academy

The Initialization Expression

```
for (var number = 0; ...; ...) {
    // Can use number here
}
// Cannot use number here
```

- Executed once, just before the loop is entered
  - Like it is out of the loop, before it
- Usually used to declare a counter variable

The Test Expression

```
for (var number = 0; number < 10; ...) {
    // Can use number here
}
// Cannot use number here
```

- Evaluated before each iteration of the loop
  - If evaluated `true`, the loop body is executed
  - If evaluated `false`, the loop body is skipped
- Used as a loop condition

The Update Expression

```
for (var number = 0; number < 10; number++) {
    // Can use number here
}
// Cannot use number here
```

- Executed at each iteration after the body of the loop is finished
- Usually used to update the counter

Telerik Academy

for Loop

Examples

Telerik Academy

Simple for Loop – Example

- A simple for-loop to print the numbers 0…9:

```
for (var number = 0; number < 10; number++) {
    console.log(number + ' ');
}
```

- A simple for-loop to calculate n!:

```
var factorial = 1;
for (var i = 1; i <= n; i++) {
    factorial *= i;
}
```

Complex for Loop – Example

- Complex for-loops could have several counter variables:

```
for (var i=1, sum=1; i<=128; i=i*2, sum+=i) {
    console.log('i=' + i + ', sum=' +sum);
}
```

Result:

```
i=1, sum=1
i=2, sum=3
i=4, sum=7
i=8, sum=15
...
```

N^M – Example

- Calculating n to power m (denoted as n^m):

```
var result = 1;

for (var i=0; i<m; i++) {
    result *= n;
}

console.log(result);
```

Telerik Academy

Calculating N^M

Live Demo

Telerik Academy

Nested Loops

Using Loops Inside a Loop

Telerik Academy

What Is Nested Loop?

- A composition of loops is called a nested loop
  - A loop inside another loop
- Example:

```
for (initialization; test; update) {
    for (initialization; test; update) {
        statements;
    }
    …
}
```

Telerik Academy

Nested Loops

Examples

Triangle – Example

- Print the following triangle:

```
1

1 2

…

1 2 3 ... n
```

```
var resultStr = '';
for(var row = 1; row <= n; row++) {
   for(var column = 1; column <= row; column++) {
      resultStr += column + ' ';
   }
   resultStr += '\n';
}
console.log(resultStr);
```

Telerik Academy

Triangle

Live Demo

Primes[N, M] – Example

- Print all prime numbers in the interval [n, m]:

```
var result='';
for (var number = n; number <= m; number++) {
    var isPrime = true;
    var divider = 2;
    var maxDivider = Math.sqrt(number);
    while (divider <= maxDivider) {
        if (number % divider == 0){
            isPrime = false;
            break;
        }
        divider++;
    }
    if (isPrime) {
        result += number + ' ';
    }
}
```

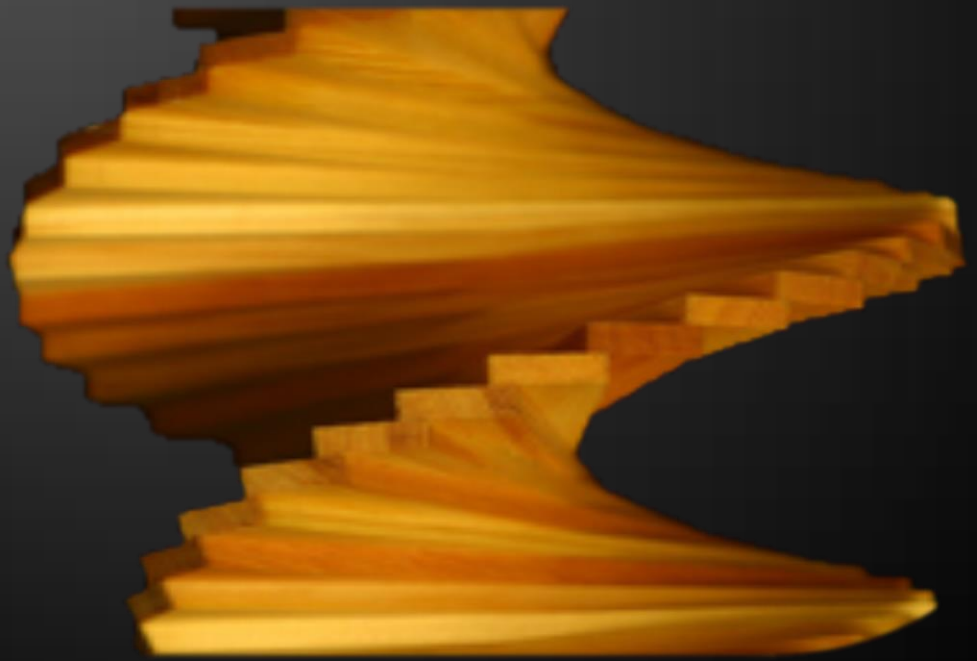
Telerik Academy

Primes in Range [n, m]

Live Demo

Telerik Academy

Loops – More Examples



Telerik Academy

Nested Loops – Examples

- Print all four digit numbers in format ABCD such that A+B = C+D (known as happy numbers)

```
for (var a =1 ; a <= 9; a++)
    for (var b = 0; b <= 9; b++
      for (var c = 0; c <= 9; c++
        for (var d = 0; d <= 9; d++)
          if (a + b == c + d)
            console.log('{0}{1}{2}{3}',
              a, b, c, d);
}
```

Can you improve this algorithm to use 3 loops only?

Telerik Academy

Nested Loops – Examples

- Print all combinations from TOTO 6/49

```
var i1, i2, i3, i4, i5, i6;
  for (i1 = 1; i1 <= 44; i1++)
    for (i2 = i1 + 1; i2 <= 45; i2+
      for (i3 = i2 + 1; i3 <= 46; i3++
        for (i4 = i3 + 1; i4 <= 47; i4++)
          for (i5 = i4 + 1; i5 <= 48; i5++)
            for (i6 = i5 + 1; i6 <= 49; i6++)
              console.log('{0} {1} {2} {3} {4} {5}',
                i1, i2, i3, i4, i5, i6);
```

Warning: execution of this code could take too long time.

Telerik Academy

for-in Loop

Iterating over the properties of object

for-in Loop

- `for-in loop` iterates over the properties of an object
  - When the object is `array`, `nodeList` or `liveNodeList` for-in iterates over their elements
  - When the object is not an array, for-in iterates over its properties

for-in Loop (2)

- Iterating over the elements of an array

```
var arr = [1, 2, 3, 4, 5, 6];
for(var index in arr) { console.log(arr[i]) }
//1, 2, 3, 4, 5, 6
```

- Iterating over the properties of document

```
for(var prop in document){ console.log(document[prop]) }
//http://localhost:64765/xxx%20for-in-loop.html
//function querySelector() { [native code] }
//function querySelectorAll() { [native code] }
//function evaluate() { [native code] }
```

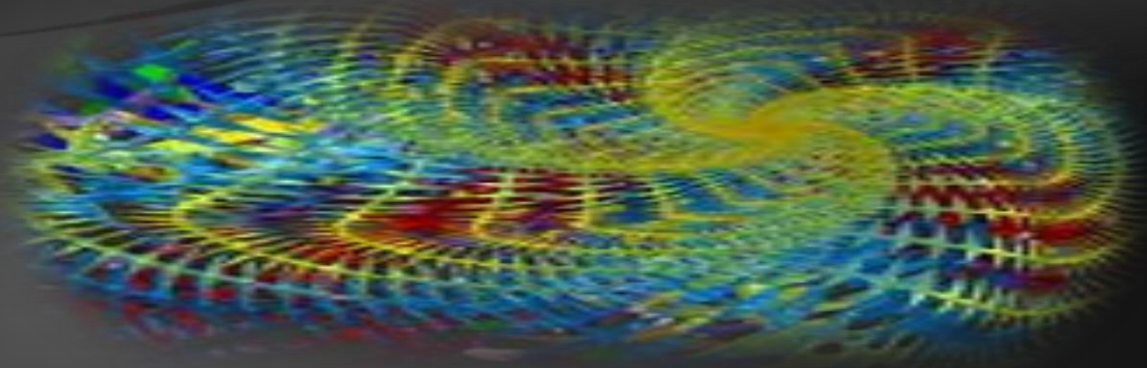
Telerik Academy

for-in Loop

Live Demo

Telerik Academy

Loops



Questions?

http://academy.telerik.com

Exercises

1. Write a script that prints all the numbers from 1 to N
2. Write a script that prints all the numbers from 1 to N, that are not divisible by 3 and 7 at the same time
3. Write a script that finds the max and min number from a sequence of numbers
4. Write a script that finds the lexicographically smallest and largest property in `document`, `window` and `navigator` objects