


FUNCTIONS

Function definition:

```
function function_name () {statement; statement;}  
function function_name (parameter, parameter){statement; statement;}
```

Function call:

```
function_name();  
function_name(argument1, argument2, ...)
```

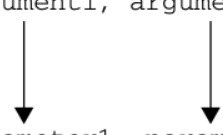


```
<html>  
  <head><title>A Simple Function</title>  
    <script type="text/javascript">  
1      function welcome(){ // Function defined within <head> tags  
2        var place="San Francisco";  
3        alert("Welcome to "+ place + "!");  
4      }  
    </script>  
  </head>
```

```
    <body bgcolor="lightblue">  
      <big>  
        <div align="center">  
          <script type="text/javascript">  
5            welcome();  
          </script>  
          <b>San Francisco</b><br />  
6            
        </div>  
      </big>  
    </body>  
  </html>
```

Pas de paramètres

```
function_name(argument1, argument2, ...); // function call (caller)  
  
function_name(parameter1, parameter2...){ // function definition (receiver)  
  var result= parameter1 + parameter2;  
  ...  
} // curly braces required
```



Crida d'una funció : des de un link

```
<html>
  <head><title>Functions</title>
1    <script type="text/javascript">
2      function greetings(){ // Function defined within <head> tags
        document.bgColor="lightblue";
        alert("Greetings to you!");
3      }
    </script>
  </head>
  <body bgcolor="silver">
    <div align="center">
4      <a href="JavaScript:greetings()"><big>Click here for
        Salutations</big>
        </a><br />
    </div>
  </body>
</html>
```

Crida d'una funció : des d'un esdeveniment

```
<html>
  <head><title>Functions and Events</title>
  <script type="text/javascript">
1    function greetings(you){ // Function definition
2      document.bgColor="lavender";
      alert("Greetings and Salutations! " + you);
    }
  </script>
</head>
```

```
3  <body>
    <div align="center">
4    <form>
5      <input type="button"
6        value="Welcome button"
7        onClick="greetings('Dan');"
        />
    </form>
    </div>
  </body>
</html>
```

Variable locals /globals

```
<html>
  <head><title>Function Scope</title>
  <script type="text/javascript">
1    var name="William";    // Global variable
    var hometown="Chico";
2    function greetme(){
3      var name="Daniel";  // Local variable
      var hometown="Husingen";
4      alert("In function the name is " + name +
        " and hometown is " + hometown);
5    }
  </script>
</head>
<body>
  <script type="text/javascript">
6    greetme();    // Function call
7    alert("Out of function, name is " + name +
8      " and hometown is " + hometown);
  </script>
</body>
</html>
```

Return de valors :

```
<html>
  <head><title>Return Value</title>
    <script type="text/javascript">
1      function mileage(miles, gas){
2        return miles/gas;    // Return the result of the division
    }
    </script>
  </head>

  <body bgcolor="lightgreen">
    <font face="arial" size="+1">
      <div align="center">
        
        <script type="text/javascript">
3          var distance=eval(prompt("How many miles did you
            drive? ", ""));
          var amount=eval(prompt("How much gas did you use?", ""));
4          var rate = mileage(distance, amount);
            // Return value assigned to rate
5          alert("Your mileage "+ rate +" miles per gallon.\n");
        </script>
      </div>
    </font>
  </body>
</html>
```

Funcions anònimes : Una funció anònima és aquella que pot ser assignada a una variable, i el valor d'aquesta serà el valor que retorni aquesta funció

```
<html>
  <head><title>Anonymous Function</title>
  <script type="text/javascript">
1      var greetings= function(){ // Anonymous function has no name
2          message="Greetings to you! "; // Function definition
3          return message;
        }
  </script>
</head>
<body>
  <big>
  <script type="text/javascript">
4      text=greetings;
        // greetings is a variable;
        // its value is the function definition
        document.write(text + "<br />");

5      text=greetings(); // Call function
        document.write(text + "<br />");
  </script>
  </big>
</body>
</html>
```

```
<html>
  <head><title>Functions as Variable</title>
  <script type="text/javascript">
1      var greetings=function (visitor){
        // Function body is assigned to greetings
        message="Greetings to you, " + visitor + "! ";
2      return message;
        }
  </script>
</head>
<body>
  <big>
  <script type="text/javascript">
3      var salutation=greetings("Elfie");
        document.write(salutation + "<br />");
4      var hello = greetings;
        // Function variable assigned to another variable
5      var welcome = greetings;
6      document.write(hello("Stranger") + "<br />");
        // Call the function
```

```

7      document.write(welcome("your Majesty") +
        " May I take your coat? </br />");
    </script>
    </big>
  </body>
</html>

```

Funcions anònimes de tancament : Son aquelles funcions anònimes que estan definides dins d'un altre funció. Quan la funció externa finalitza, retorna una referència a la funció anònima interior pel qual és possible cridar-la a través d'aquesta referència fent que les variables locals siguin accessibles encara que la funció externa hagi finalitzat.

Un ús comú d'aquest tipus de funcions son per configurar els paràmetres per a una funció que serà cridada en algun moment del futur

```

<html>
  <head><title>Closures</title>
    <script type="text/javascript">
1      function paint(type, color) {
2        var str = "The " + type + " is " + color; //local variable
3        var tellme = function() { // Anonymous function
          document.write("<big>" + str + "</big><br />")
        }
4      return tellme; // return a reference to the function
    }
    </script>
  </head>
  <body bgColor="lightgreen">
    <h2>Testing a closure</h2>
    <script type="text/javascript">
5      var say1 = paint("rose", "red"); /* A reference to the
                                           anonymous is
                                           function is returned */
6      var say2 = paint("sun", "yellow");
7      alert(say1); // See the value in say1 Figure 7.13
8      say1(); // Call the anonymous function
9      say2();
    </script>
  </body>
</html>

```

Exercicis :

1. Realitzar una funció recursiva de la sèrie de Fibonacci : 0,1,1,2,3,5,8,13,21,34,55
2. Amb la funció recursiva anterior l'usuari podrà demanar en quin nombre a partir del qual vol que comenci la serie i en quin nombre ha d'acabar
3. Realitzar una funció anònima que calculi i retorni la distància que has recorregut sabent el temps i la velocitat. Aquesta funció s'ha d'assignar a una variable
4. Dissenyeu una programa amb Javascript que calculi el preu de l'estada dins d'un pàrquing. A de tindre una funció demani tant l'hora d'entrada com la sortida en format hora i minut (dos enters), els quals suposem que són del mateix dia. Per calcular el preu cridarem a una altre funció sabent que ens costa 1 euro cada hora o fracció, excepte la primera hora, que la subvenciona l'associació de botiguers.