

JAVASCRIPT

JavaScript

Η **JavaScript** είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που χρησιμοποιείται για να προσθέσει εφέ και διαλογικότητα (αλληλεπίδραση, διαδραστικότητα) στις ιστοσελίδες μας.

Ο κώδικας της **JavaScript** περιέχεται ανάμεσα στα tags **<script>** και **</script>** και σαν χαρακτηριστικό (attribute) μπορούμε να χρησιμοποιήσουμε το `type="text/javascript"` ή το `language="JavaScript"`. Στην καινούρια έκδοση της JavaScript δεν χρειάζεται το `type`.

Μια **συνάρτηση (function)** JavaScript είναι ένα μπλοκ του κώδικα JavaScript, που μπορεί να εκτελεστεί όταν την καλούμε.

Το script μπορεί να είναι στο **<head>** αλλά και στο **<body>**.

Μια συνάρτηση (function) (που είναι μέσα σε ένα script) τοποθετείται στο **<head>**

Ενσωμάτωση κώδικα JavaScript σε ένα πρόγραμμα html:

- **Internal JavaScript:** Εισάγουμε με την ετικέτα **script** ένα κώδικα, σε οποιοδήποτε μέρος του εγγράφου όμως το πιο συνηθισμένο είναι να γίνεται στην περιοχή head στην επικεφαλίδα του εγγράφου html.

```
<script>
.....
</script>
```
- **External JavaScript:** Χρησιμοποιήσουμε την ετικέτα **script** εισάγοντας ένα σύνδεσμο σε ένα **εξωτερικό JavaScript αρχείο** με κατάληξη **.js**.

```
<script src="myScript.js"></script>
```
- **Inline script:** παίρνει κάποια εντολή JavaScript σαν αποτέλεσμα κάποιου γεγονότος μαζί με τον ορισμό μιας ετικέτας σε ένα στοιχείο.

```
<button onclick="document.write('Hello')">click</button>
```

Εντολή	Επεξήγηση
Εμφάνιση δεδομένων	Γράφοντας σε ένα στοιχείο HTML, χρησιμοποιώντας το innerHTML . Γράφοντας στην έξοδο HTML χρησιμοποιώντας το document.write () . Γράφοντας σε ένα πλαίσιο ειδοποίησης, χρησιμοποιώντας το window.alert () . Γράφοντας στην κονσόλα του προγράμματος περιήγησης, χρησιμοποιώντας το console.log () .
document.write()	Εμφανίζει κείμενο στην οθόνη του φυλλομετρητή: <pre><script> document.write("Hello from JavaScript") </script></pre>
innerHTML	Για να έχουμε πρόσβαση σε ένα στοιχείο HTML, η JavaScript μπορεί να χρησιμοποιήσει τη μέθοδο document.getElementById(id) . Το χαρακτηριστικό id καθορίζει το στοιχείο HTML. Η ιδιότητα innerHTML ορίζει το περιεχόμενο HTML.
getElementById()	Χρησιμοποιεί το id ενός στοιχείου HTML για να αλλάξει το περιεχόμενο :

	<p>Παράδειγμα: χρησιμοποιώντας το id της παραγράφου, αλλάζει το περιεχόμενο (innerHTML) της.</p> <pre><p id="demo">..... JavaScript!</p></pre> <pre><button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!'">Click Me!</button></pre>
	<p>Χρησιμοποιεί το id ενός στοιχείου HTML για να αλλάξει μια ιδιότητα:</p> <pre><button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the light</button></pre> <pre></pre> <pre><button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the light</button></pre>
	<p>Χρησιμοποιεί το id ενός στοιχείου HTML για να αλλάξει το style css:</p> <pre>document.getElementById('demo').style.fontSize='35px'</pre>
	<p>Χρησιμοποιεί το id ενός στοιχείου HTML για να κρύψει ένα html στοιχείο με την ιδιότητα display ή γενικά να αλλάξει την ιδιότητα display:</p> <pre>document.getElementById("demo").style.display = "none";</pre> <pre><p id="demo" style="display:none">Hello JavaScript!</p></pre> <pre><button type="button" onclick="document.getElementById('demo').style.display='block'">Click Me!</button></pre>
window.alert()	<p>ένα πλαίσιο ειδοποίησης για την εμφάνιση δεδομένων.</p> <pre>window.alert(5 + 6);</pre> <pre>alert("Hello");</pre>
console.log ()	<p>Για σκοπούς εντοπισμού σφαλμάτων, μπορείτε να καλέσετε τη μέθοδο console.log () στο πρόγραμμα περιήγησης για να εμφανίσετε δεδομένα.</p> <pre>console.log(x);</pre>
window.print()	<p>Μπορείτε να καλέσετε τη μέθοδο window.print () στο πρόγραμμα περιήγησης για να εκτυπώσετε το περιεχόμενο του τρέχοντος παραθύρου.</p>

Εντολή	Επεξήγηση
Μεταβλητές	<p>Σε μια γλώσσα προγραμματισμού, οι μεταβλητές χρησιμοποιούνται για την αποθήκευση τιμών δεδομένων.</p> <p>Χρησιμοποιεί τη λέξη-κλειδί var για να δηλώσει μεταβλητές. Από την έκδοση 6: let</p> <pre>var x;</pre> <pre>var pi = 3.14;</pre> <pre>var name = "anna";</pre> <pre>x = 5;</pre>
	αριθμητικούς τελεστές (+ - * / % ++ -- **) για τον υπολογισμό τιμών

	<pre>x = 5 + 6; x += 10; x++; y = x * 10; name = "anna" + " " + "pap";</pre>	
	Τύποι δεδομένων: numbers, strings, objects, boolean	
Σχόλια	// μια γραμμή /* πολλές γραμμές */	
Τελεστές σύγκρισης	Τελεστής	Περιγραφή
	==	Ίσον
	===	Ίσον τιμή και τύπος
	!=	Όχι ίσον
	!==	Όχι ίσον τιμή και τύπος
	>	Μεγαλύτερο από
	<	Μικρότερο από
	>=	Μεγαλύτερο από ή ίσον
	<=	Μικρότερο από ή ίσον
	?	τριαδικός χειριστής
Λογικοί Τελεστές	&&	and
		or
	!	not

Εντολή	Επεξήγηση
String	Οι συμβολοσειρές JavaScript χρησιμοποιούνται για την αποθήκευση και το χειρισμό κειμένου. <pre>var fname='anna'; var lname="pap"; fname = fname + lname;</pre>
String Length	<pre>var txt = "good morning"; var sln = txt.length;</pre>
indexOf()	επιστρέφει τη θέση της πρώτης εμφάνισης ενός καθορισμένου κειμένου σε μια συμβολοσειρά: <pre>var str = "please visit microsoft and microsoft "; var pos = str.indexOf("microsoft "); var pos = str.indexOf("microsoft ", 15);</pre>
lastIndexOf()	επιστρέφει το ευρετήριο της τελευταίας εμφάνισης ενός καθορισμένου κειμένου σε μια συμβολοσειρά:
search()	αναζητά μια συμβολοσειρά για μια καθορισμένη τιμή και επιστρέφει τη θέση: <pre>var pos = str.search("microsoft ");</pre>
slice()	εξάγει ένα μέρος μιας συμβολοσειράς και επιστρέφει το εξαγόμενο τμήμα σε μια νέα συμβολοσειρά. Η μέθοδος παίρνει 2 παραμέτρους: τη θέση έναρξης και τη θέση τερματισμού (δεν περιλαμβάνεται το τέλος). <pre>var str = "Apple, Banana, Kiwi"; var res = str.slice(7, 13);</pre> substring(),substr(): παρόμοια
replace()	αντικαθιστά μια καθορισμένη τιμή με μια άλλη τιμή σε μια συμβολοσειρά: <pre>str = "Please visit Microsoft!"; var n = str.replace("Microsoft", "W3Schools");</pre>
toUpperCase() toLowerCase()	Μια συμβολοσειρά μετατρέπεται σε κεφαλαία/πεζά
split()	Μια συμβολοσειρά μπορεί να μετατραπεί σε πίνακα: <pre>var txt = "a,b,c,d,e"; txt.split(",");</pre>

Εντολή	Επεξήγηση
Number	Η JavaScript έχει μόνο έναν τύπο αριθμού. Οι αριθμοί μπορούν να γραφτούν με ή χωρίς δεκαδικά ψηφία. <pre>var x = 3.14; var y = 3; var x = 10; var y = 20; var z = x + y; // z will be 30 (a number) var x = 10; var y = "20"; var z = x + y; // z will be 1020 (a string)</pre>

	<pre>var x = "100"; var y = "10"; var z = x / y; // z will be 10 var z = x * y; // z will be 1000</pre>
NaN - Not a Number	<p>Το NaN είναι μια δεσμευμένη JavaScript λέξη που δείχνει ότι αν μια μεταβλητή δεν είναι ένας νόμιμος αριθμός.</p> <pre>var x = 100 / "Apple"; isNaN(x); // returns true because x is Not a Number</pre>
Infinity	<p>Infinity (ή -Infinity) είναι η τιμή που θα επιστρέψει η JavaScript εάν υπολογίσετε έναν αριθμό εκτός του μεγαλύτερου δυνατού αριθμού.</p> <pre>var x = 2 / 0; // x will be Infinity var y = -2 / 0; // y will be -Infinity</pre>
toString()	<p>επιστρέφει έναν αριθμό ως συμβολοσειρά.</p> <pre>var x = 123; x.toString();</pre>
toFixed()	<p>επιστρέφει μια συμβολοσειρά, με τον αριθμό γραμμένο με έναν καθορισμένο αριθμό δεκαδικών:</p> <pre>var x = 9.656; x.toFixed(0); // returns 10 x.toFixed(2); // returns 9.66</pre>
Number()	<p>μπορεί να χρησιμοποιηθεί για τη μετατροπή μεταβλητών JavaScript σε αριθμούς:</p> <pre>Number(true); // returns 1 Number(false); // returns 0 Number("10"); // returns 10 Number("10.33"); // returns 10.33 Number("10,33"); // returns NaN</pre>
parseInt()	<p>αναλύει μια συμβολοσειρά και επιστρέφει έναν ακέραιο αριθμό.</p> <pre>parseInt("10"); // returns 10 parseInt("10.33"); // returns 10 parseInt("10 20 30"); // returns 10 parseInt("10 years"); // returns 10 parseInt("years 10"); // returns NaN</pre>
parseFloat()	<p>αναλύει μια συμβολοσειρά και επιστρέφει έναν αριθμό.</p> <pre>parseFloat("10"); // returns 10 parseFloat("10.33"); // returns 10.33</pre>
Math.pow(x, y)	επιστρέφει την τιμή του x υψωμένη στην y
Math.sqrt(x)	επιστρέφει την τετραγωνική ρίζα του x: <pre>Math.sqrt(64);</pre>
Math.abs(x)	επιστρέφει την απόλυτη (θετική) τιμή του x.
Math.round()	επιστρέφει την τιμή του x στρογγυλεμένη στον πλησιέστερο ακέραιο αριθμό: <pre>Math.round(4.7); // returns 5 Math.round(4.4); // returns 4</pre>
Math.ceil(x)	επιστρέφει την τιμή του x στρογγυλεμένη προς τα πάνω στον πλησιέστερο ακέραιο αριθμό: <pre>Math.ceil(4.4); // returns 5</pre>
Math.floor(x)	επιστρέφει την τιμή του x στρογγυλεμένη προς τα κάτω στον πλησιέστερο

	ακέραιο αριθμό: <code>Math.floor(4.7); // returns 4</code>
Math.min(), Math.max()	εύρεση της χαμηλότερης ή υψηλότερης τιμής σε μια λίστα ορισμάτων: <code>Math.max(0, 150, 30, 20, -8, -200); // returns 150</code>
Math.random()	επιστρέφει έναν τυχαίο αριθμό μεταξύ 0 (συμπεριλαμβανομένου) και 1 (όχι το 1): <code>Math.random();</code> Το <code>Math.random ()</code> που χρησιμοποιείται με το <code>Math.floor ()</code> μπορεί να χρησιμοποιηθεί για την επιστροφή τυχαίων ακέραιων αριθμών: <code>Math.floor(Math.random() * 10); // returns a random integer from 0 to 9</code> <code>Math.floor(Math.random() * 10) + 1; // returns a random integer from 1 to 10</code>

Εντολή	Επεξήγηση
Arrays	Οι πίνακες JavaScript χρησιμοποιούνται για την αποθήκευση πολλαπλών τιμών σε μία μόνο μεταβλητή. <code>var array_name = [item1, item2, ...];</code> <code>var cars = ["Saab", "Volvo", "BMW"];</code>
length	επιστρέφει το μήκος ενός πίνακα (ο αριθμός των στοιχείων πίνακα). <code>var fruits = ["Banana", "Orange", "Apple", "Mango"];</code> <code>fruits.length; // the length of fruits is 4</code> <code>fLen = fruits.length;</code> <code>text = "";</code> <code>for (i = 0; i < fLen; i++) {</code> <code> text += "" + fruits[i] + "";</code> <code>}</code> <code>text += "";</code>
push()	προσθήκη ενός νέου στοιχείου σε έναν πίνακα στο τέλος. <code>fruits.push("Lemon");</code> Νέο στοιχείο μπορεί επίσης να προστεθεί σε έναν πίνακα χρησιμοποιώντας την ιδιότητα <code>length</code> : <code>fruits[fruits.length] = "Lemon";</code> <code>var x = fruits.push("Kiwi"); // επιστρέφει το πλήθος</code>
pop()	αφαιρεί το τελευταίο στοιχείο από έναν πίνακα: <code>fruits.pop();</code> <code>var x = fruits.pop(); // επιστρέφει το στοιχείο που διαγράφηκε</code>
shift()	αφαιρεί το πρώτο στοιχείο πίνακα και "μετατοπίζει" όλα τα άλλα στοιχεία. <code>fruits.shift();</code>
unshift()	προσθέτει ένα νέο στοιχείο σε έναν πίνακα (στην αρχή). <code>fruits.unshift("Lemon");</code>
delete	Δεδομένου ότι οι πίνακες JavaScript είναι αντικείμενα, τα στοιχεία μπορούν να διαγραφούν με το <code>delete</code> : <code>delete fruits[0];</code>

splice()	<p>μπορεί να χρησιμοποιηθεί για την προσθήκη νέων στοιχείων σε έναν πίνακα: <code>fruits.splice(2, 0, "Lemon", "Kiwi");</code></p> <p>Η πρώτη παράμετρος (2) καθορίζει τη θέση στην οποία πρέπει να προστεθούν νέα στοιχεία. Η δεύτερη παράμετρος (0) καθορίζει πόσα στοιχεία πρέπει να αφαιρεθούν. Οι υπόλοιπες παράμετροι καθορίζουν τα νέα στοιχεία που θα προστεθούν.</p> <p>μπορείτε να χρησιμοποιήσετε το <code>splice ()</code> για να αφαιρέσετε στοιχεία: <code>fruits.splice(0, 1); // διαγράφει το 1ο</code></p>
concat ()	<p>Η μέθοδος <code>concat ()</code> δημιουργεί έναν νέο πίνακα <code>var myGirls = ["Cecilie", "Lone"];</code> <code>var myBoys = ["Emil", "Tobias", "Linus"];</code> <code>var myChildren = myGirls.concat(myBoys);</code></p>
slice()	<p>κόβει ένα κομμάτι ενός πίνακα σε έναν νέο πίνακα: <code>var citrus = fruits.slice(1);</code> <code>var citrus = fruits.slice(1, 3);</code></p>
toString()	<p>μετατρέπει έναν πίνακα σε μια συμβολοσειρά (διαχωρισμένη με κόμμα). <code>x = fruits.toString();</code></p>
join()	<p>Η μέθοδος ενώνει επίσης όλα τα στοιχεία πίνακα σε μια συμβολοσειρά. Συμπεριφέρεται όπως η <code>toString ()</code>, αλλά επιπλέον μπορείτε να καθορίσετε το διαχωριστικό: <code>x = fruits.join(" * ");</code></p>
sort()	<p>ταξινομεί έναν πίνακα αλφαβητικά: <code>fruits.sort();</code></p>
reverse()	<p>αντιστρέφει τα στοιχεία σε έναν πίνακα. Μπορείτε να το χρησιμοποιήσετε για να ταξινομήσετε έναν πίνακα σε φθίνουσα σειρά: <code>fruits.reverse();</code></p>
indexOf ()	<p>αναζητά σε έναν πίνακα για μια τιμή στοιχείου και επιστρέφει τη θέση του. <code>var a = fruits.indexOf("Apple");</code></p>

Εντολή	Επεξήγηση
new Date()	<p>δημιουργεί ένα νέο αντικείμενο ημερομηνίας με την τρέχουσα ημερομηνία και ώρα: <code>var d = new Date();</code></p>
getFullYear()	<p>επιστρέφει το έτος μιας ημερομηνίας ως τετραψήφιο αριθμό: <code>var d = new Date();</code> <code>document.getElementById("demo").innerHTML = d.getFullYear();</code></p> <p>αντίστοιχα είναι: <code>getMonth()</code>,<code>getDate()</code>,</p>
setFullYear()	<p>ορίζει το έτος ενός αντικειμένου ημερομηνίας. <code><script></code> <code>var d = new Date();</code> <code>d.setFullYear(2020);</code></p> <p><code>setMonth()</code>,<code>setDate()</code>.</p>

Εντολή	Επεξήγηση
if	<p>Χρησιμοποιήστε τη δήλωση if για να καθορίσετε ένα μπλοκ κώδικα JavaScript που θα εκτελεστεί εάν μια συνθήκη είναι αληθής.</p> <pre> if (condition) { // block of code to be executed if the condition is true } if (condition) { // block of code to be executed if the condition is true } else { // block of code to be executed if the condition is false } if (condition1) { // block of code to be executed if condition1 is true } else if (condition2) { // block of code to be executed if the condition1 is false and condition2 is true } else { // block of code to be executed if the condition1 is false and condition2 is false } if (hour < 18) { greeting = "Good day"; } else { greeting = "Good evening"; } variablename = (condition) ? value1:value2 var voteable = (age < 18) ? "young":"Old "; </pre>
switch	<p>χρησιμοποιείται για την εκτέλεση διαφορετικών ενεργειών βάσει διαφορετικών συνθηκών.</p> <pre> switch(expression) { case x: // code block break; case y: // code block break; default: // code block } </pre>
for	<p>Οι βρόχοι μπορούν να εκτελέσουν ένα μπλοκ κώδικα πολλές φορές.</p> <pre> for (statement 1; statement 2; statement 3) { // code block to be executed } </pre>

	<p>statement 1: εκτελείται (μία φορά) πριν από την εκτέλεση του μπλοκ κώδικα.</p> <p>statement 2: ορίζει την προϋπόθεση για την εκτέλεση του μπλοκ κώδικα. Αν ορίσουμε μία συνθήκη τότε εκτελείται όσο είναι αληθής η συνθήκη.</p> <p>statement 3: εκτελείται (κάθε φορά) μετά την εκτέλεση του μπλοκ κώδικα. Είναι το βήμα.</p> <pre>for (i = 0; i < 5; i++) { text += "The number is " + i + "
"; }</pre> <p>for/in:</p> <pre>var cars = ['BMW', 'Volvo', 'Mini']; for (x in cars) { text += cars[x]; }</pre> <p>for/of (περιβάλλει τις τιμές ενός επαναλαμβανόμενου αντικειμένου):</p> <pre>var cars = ['BMW', 'Volvo', 'Mini']; for (x of cars) { document.write(x + "
"); }</pre>
while	<p>Ο βρόχος while εκτελεί ένα μπλοκ κώδικα, εφόσον ισχύει μια συγκεκριμένη συνθήκη.</p> <pre>while (condition) { // code block to be executed }</pre> <pre>while (i < 10) { text += "The number is " + i; i++; }</pre> <pre>do { text += "The number is " + i; i++; }while (i < 10);</pre>
break	Φεύγει από έναν βρόχο.
continue	Πηγαίνει στην επόμενη επανάληψη.

Εντολή	Επεξήγηση
function	<p>είναι ένα μπλοκ κώδικα που έχει σχεδιαστεί για να εκτελεί μια συγκεκριμένη εργασία.</p> <pre>function name(parameter1, parameter2, parameter3) { // code to be executed }</pre>

	<pre> } var x = myFunction(4, 3); function myFunction(a, b) { return a * b; } x = sumAll(1, 123, 500, 115, 44, 88); function sumAll() { var i; var sum = 0; for (i = 0; i < arguments.length; i++) { sum += arguments[i]; } return sum; } </pre>														
events	<p>Τα συμβάντα HTML είναι "πράγματα" που συμβαίνουν σε στοιχεία HTML.</p> <pre> <element event='some JavaScript'> <button onclick="this.innerHTML= Date()">The time is?</button> <button onclick="displayDate()">The time is?</button> </pre>														
	<table border="1"> <thead> <tr> <th data-bbox="368 1149 608 1249">Event</th> <th data-bbox="608 1149 1437 1249">περιγραφή</th> </tr> </thead> <tbody> <tr> <td data-bbox="368 1249 608 1361">onchange</td> <td data-bbox="608 1249 1437 1361">Ένα στοιχείο HTML έχει αλλάξει</td> </tr> <tr> <td data-bbox="368 1361 608 1462">onclick</td> <td data-bbox="608 1361 1437 1462">Ο χρήστης κάνει κλικ σε ένα στοιχείο HTML</td> </tr> <tr> <td data-bbox="368 1462 608 1574">onmouseover</td> <td data-bbox="608 1462 1437 1574">Ο χρήστης μετακινεί το ποντίκι πάνω από ένα στοιχείο HTML</td> </tr> <tr> <td data-bbox="368 1574 608 1686">onmouseout</td> <td data-bbox="608 1574 1437 1686">Ο χρήστης απομακρύνει το ποντίκι από ένα στοιχείο HTML</td> </tr> <tr> <td data-bbox="368 1686 608 1798">onkeydown</td> <td data-bbox="608 1686 1437 1798">Ο χρήστης πιέζει ένα πλήκτρο πληκτρολογίου</td> </tr> <tr> <td data-bbox="368 1798 608 1883">onload</td> <td data-bbox="608 1798 1437 1883">Το πρόγραμμα περιήγησης έχει ολοκληρώσει τη φόρτωση</td> </tr> </tbody> </table>	Event	περιγραφή	onchange	Ένα στοιχείο HTML έχει αλλάξει	onclick	Ο χρήστης κάνει κλικ σε ένα στοιχείο HTML	onmouseover	Ο χρήστης μετακινεί το ποντίκι πάνω από ένα στοιχείο HTML	onmouseout	Ο χρήστης απομακρύνει το ποντίκι από ένα στοιχείο HTML	onkeydown	Ο χρήστης πιέζει ένα πλήκτρο πληκτρολογίου	onload	Το πρόγραμμα περιήγησης έχει ολοκληρώσει τη φόρτωση
Event	περιγραφή														
onchange	Ένα στοιχείο HTML έχει αλλάξει														
onclick	Ο χρήστης κάνει κλικ σε ένα στοιχείο HTML														
onmouseover	Ο χρήστης μετακινεί το ποντίκι πάνω από ένα στοιχείο HTML														
onmouseout	Ο χρήστης απομακρύνει το ποντίκι από ένα στοιχείο HTML														
onkeydown	Ο χρήστης πιέζει ένα πλήκτρο πληκτρολογίου														
onload	Το πρόγραμμα περιήγησης έχει ολοκληρώσει τη φόρτωση														

Εντολή	Επεξήγηση
Form	<p>Η επικύρωση φόρμας HTML μπορεί να γίνει με JavaScript.</p> <p>Εάν ένα πεδίο φόρμας (fname) είναι κενό, αυτή η συνάρτηση ειδοποιεί ένα μήνυμα και επιστρέφει false, για να αποτρέψει την υποβολή της φόρμας:</p> <pre>function validateForm() { var x = document.forms["myForm"]["fname"].value; if (x == "") { alert("Name must be filled out"); return false; } }</pre> <p><form name="myForm" action="/action_page.php" onsubmit="return validateForm()" method="post"></p>

Εντολή	Επεξήγηση
"use strict";	Ορίζει ότι ο κώδικας JavaScript πρέπει να εκτελεστεί σε "αυστηρή λειτουργία".

Εντολή	Επεξήγηση
Errors	<p>Όταν παρουσιαστεί σφάλμα, η JavaScript κανονικά θα σταματήσει και θα δημιουργήσει ένα μήνυμα σφάλματος.</p> <p>Try: επιτρέπει να δοκιμάσετε ένα μπλοκ κώδικα για σφάλματα.</p> <p>Catch: επιτρέπει να χειριστείτε το σφάλμα.</p> <p>Throw: επιτρέπει να δημιουργείτε προσαρμοσμένα σφάλματα.</p> <p>Finally: επιτρέπει να εκτελέσετε κώδικα ανεξάρτητα από το αποτέλεσμα.</p> <pre>try { Block of code to try } catch(err) { Block of code to handle errors }</pre> <p><p id="demo"></p></p> <pre><script> try { adddler("Welcome guest!"); } catch(err) { document.getElementById("demo").innerHTML = err.message; } </script></pre>

```

try {
  if(x == "") throw "is empty";
  if(isNaN(x)) throw "is not a number";
  x = Number(x);
  if(x > 10) throw "is too high";
  if(x < 5) throw "is too low";
}
catch(err) {
  message.innerHTML = "Error: " + err + ".";
  message.innerHTML = err.name;
}
finally {
  document.getElementById("demo").value = "";
}

```

HTML DOM (Document Object Model)

