



JavaScript™

Τεχνολογίες προγραμματισμού στη
πλευρά του πελάτη
client side scripting technologies

Document **O**bject **M**odel

Αντικείμενα - Objects



Αντικείμενα - Objects

Ιδιότητες - Properties

Γεγονότα - Events

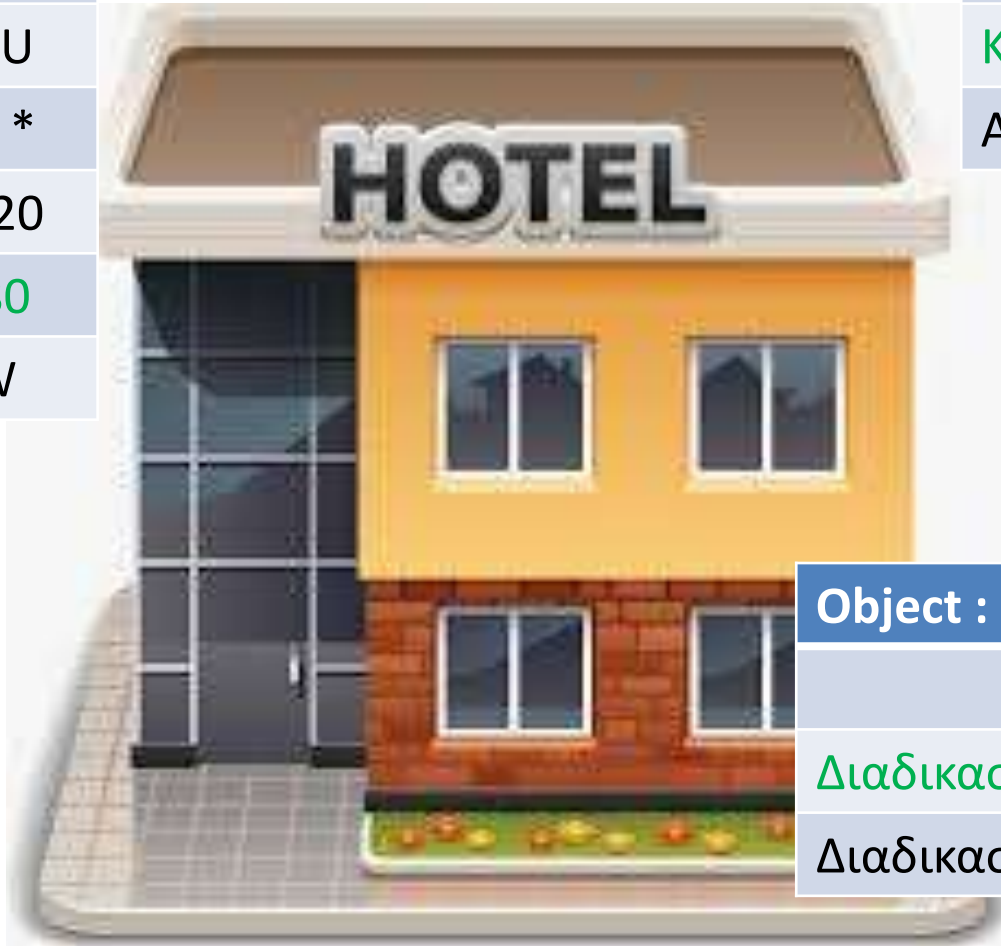
Μεθόδους - Methods



Αντικείμενα - Objects

Ιδιότητες - Γεγονότα – Μέθοδοι (Properties - Events- Methods)

Object : Hotel	
Properties	
Όνομα	4U
Κατηγορία	5 *
Δωμάτια	120
Κρατήσεις	80
Πισίνα	ν



Object : Hotel
Event
Κράτηση
Ακύρωση

Object : Hotel
Method
Διαδικασία κράτησης
Διαδικασία ακύρωσης

Αντικείμενα - Objects

Ιδιότητες - Γεγονότα –Μέθοδοι (Properties - Events- Methods)

Object : Car

Properties

Μάρκα	Audi
Χρώμα	Κόκκινο
Καύσιμα	Βενζίνη
Ταχύτητα	80

Object : Car

Event

Ο οδηγός πατά γκάζι
Ο οδηγός πατά φρένο

Object : Car

Method

Επιτάχυνση
Επιβράδυνση

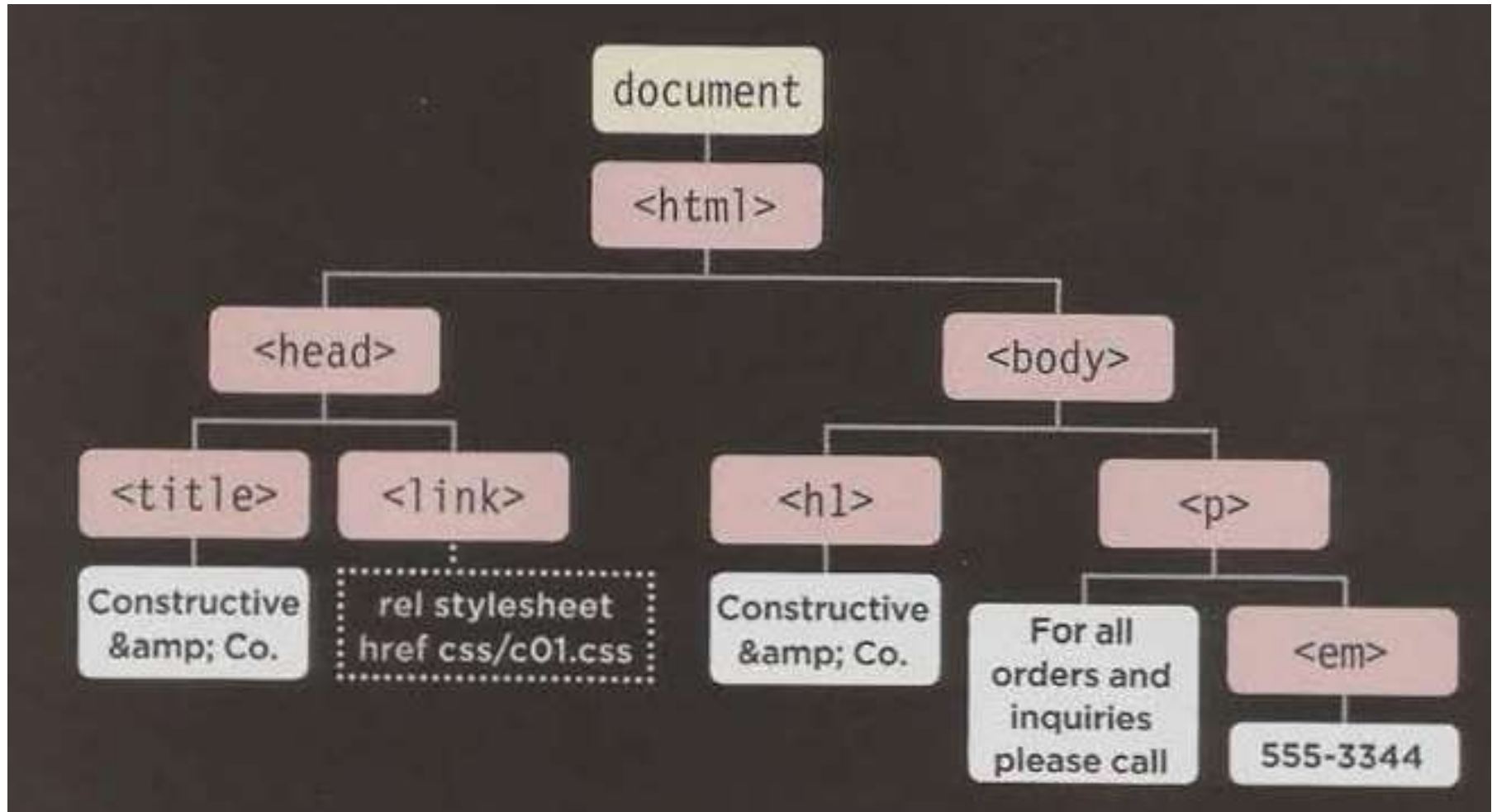


Αντικείμενα στην Javascript

- Στον προγραμματισμό τα αντικείμενα ομαδοποιούν ένα σύνολο μεταβλητών και συναρτήσεων για να δημιουργήσουν ένα μοντέλο που είναι αναγνωρίσιμο από τον πραγματικό κόσμο.
- Σε ένα αντικείμενο οι μεταβλητές ονομάζονται Ιδιότητες. Μας δίνουν πληροφορίες για το αντικείμενο (αριθμός δωματίων)
- Σε ένα αντικείμενο οι συναρτήσεις ονομάζονται Μέθοδοι. Αντιπροσωπεύουν τις εργασίες που σχετίζονται με το αντικείμενο (έλεγχος διαθεσιμότητας δωματίου)

ΑΝΤΙΚΕΙΜΕΝΑ - Objects

Οι Web browsers κτίζουν μια σελίδα από διάφορα αντικείμενα



ΑΝΤΙΚΕΙΜΕΝΑ στην Javascript

Αυτό το αντικείμενο αντιπροσωπεύει ένα ξενοδοχείο. Έχει πέντε ιδιότητες και μία μέθοδο. Αποθηκεύεται σε μια μεταβλητή που ονομάζεται ξενοδοχείο.

```
var hotel = {
```

```
  name: 'Quay',
```

```
  rooms: 40,
```

```
  booked: 25,
```

```
  gym: true,
```

```
  roomTypes: ['twin', 'double', 'suite'],
```

```
  checkAvailability: function() {  
    return this.rooms - this.booked;  
  }  
}
```

```
};
```

● KEY
● VALUE

PROPERTIES
These are variables

METHOD
This is a function

ΑΝΤΙΚΕΙΜΕΝΑ στην Javascript

Αυτό το αντικείμενο αντιπροσωπεύει ένα ξενοδοχείο. Έχει πέντε ιδιότητες και μία μέθοδο. Αποθηκεύεται σε μια μεταβλητή που ονομάζεται ξενοδοχείο.

```
var hotel = {
```

● KEY
● VALUE

```
  name: 'Quay',
```

```
  rooms: 40,
```

```
  booked: 25,
```

```
  gym: true,
```

```
  roomTypes: ['twin', 'double', 'suite'],
```

PROPERTIES

These are variables

```
  checkAvailability: function() {
```

```
    return this.rooms - this.booked;
```

```
  }
```

METHOD

This is a function

```
};
```

Ιδιότητες Αντικειμένων στη Javascript

// Άμεσος ορισμός αντικειμένου

<script>

```
emp={id:102,name:"Γιάννης Άνυφτος",salary:40000}  
document.write(emp.id+" "+emp.name+" "+emp.salary);
```

</script>

<script>

```
var emp=new Object();  
emp.id=101;  
emp.name=«Κώστας Αχαίρευτος»;  
emp.salary=50000;  
document.write(emp.id+" "+emp.name+" "+emp.salary);
```

</script>

ΑΝΤΙΚΕΙΜΕΝΑ στην Javascript

- Κλίση Ιδιότητας ή Μεθόδου:

```
Var hName = hotel.name;
```

```
Var fRooms = hotel.checkAvailability();
```

- Αλλαγή τιμής ιδιότητας:

```
hotel.name = "4U"           ή
```

```
hotel['name'] = "4U"
```

- Διαγραφή τιμής / ιδιότητας:

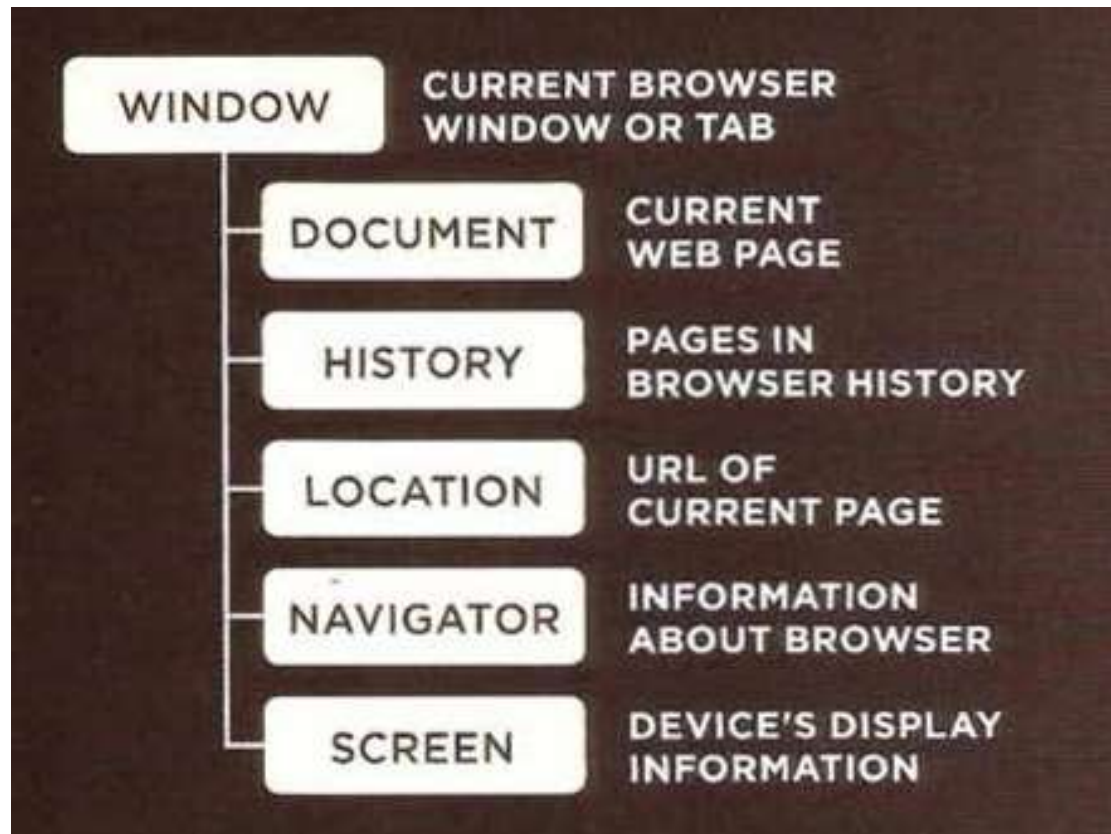
```
hotel.name = ""
```

```
Delete hotel.name
```

Διαθέσιμα Αντικείμενα στην Javascript

- **BROWSER OBJECT MODEL**

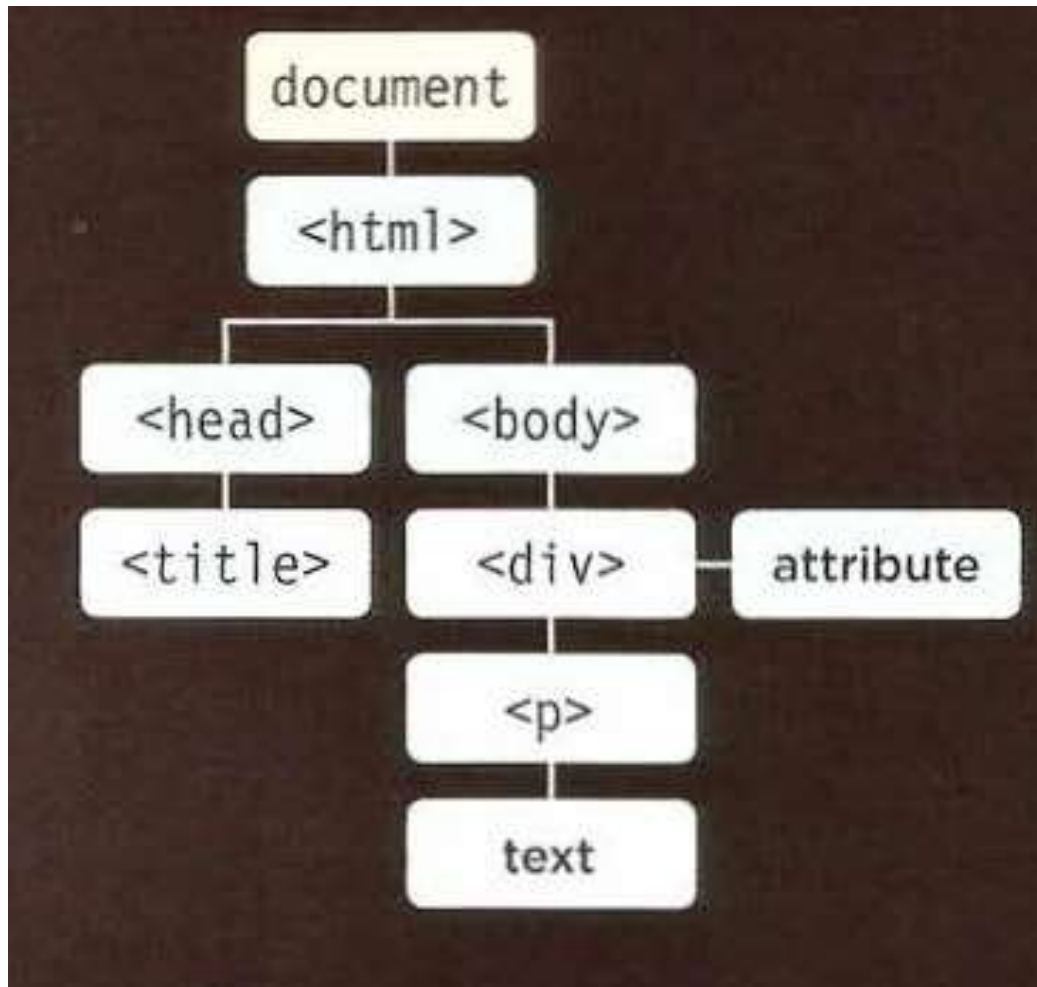
- Περιέχει αντικείμενα που αντιπροσωπεύουν το τρέχον παράθυρο του Browser. (ιστορικό είδος οθόνης της συσκευής)



Διαθέσιμα Αντικείμενα στην Javascript

- **DOCUMENT OBJECT MODEL**

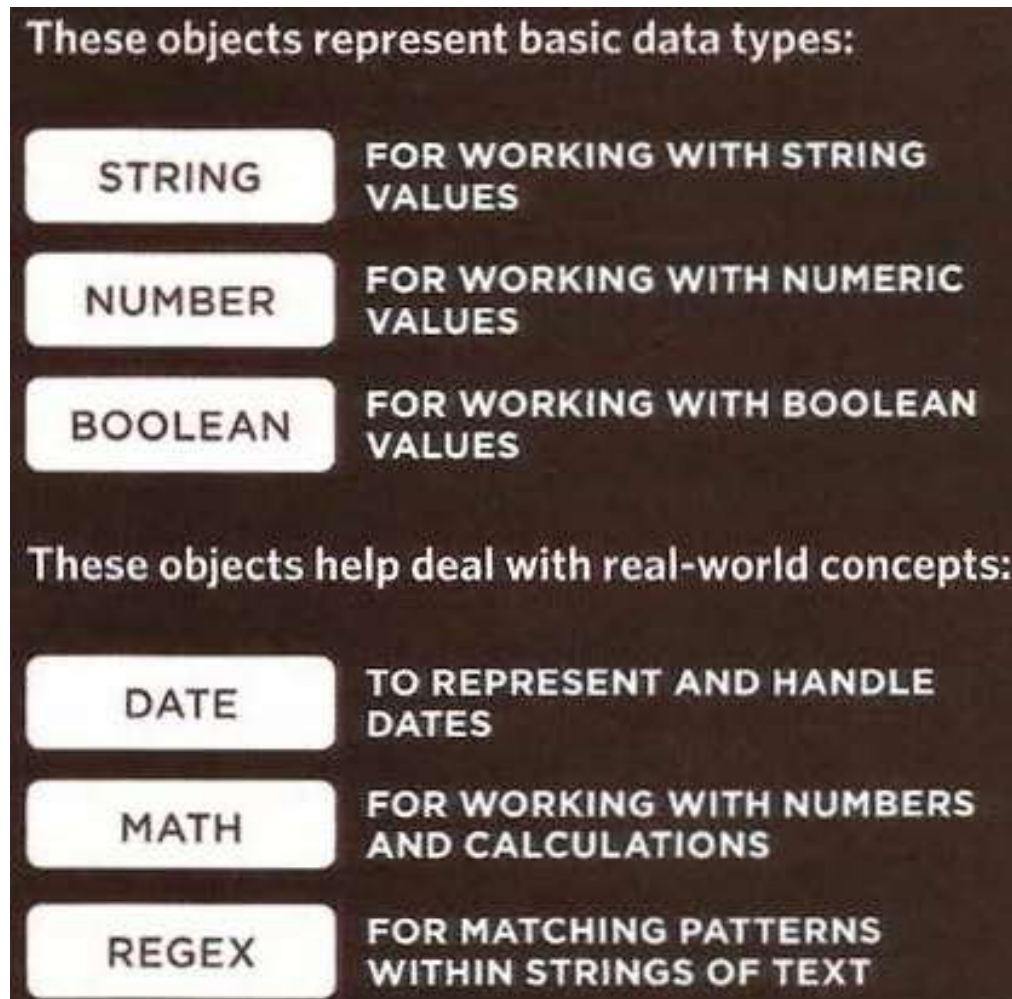
- Δημιουργεί ένα αντικείμενο για κάθε στοιχείο της ιστοσελίδας.



Διαθέσιμα Αντικείμενα στην Javascript

- **GLOBAL JAVASCRIPT OBJECTS**

- Περιέχει ενσωματωμένα αντικείμενα που χρησιμεύουν στο κτίσιμο της σελίδας. Πχ Date()



THE BROWSER OBJECT MODEL: THE WINDOW OBJECT

Αντιπροσωπεύει το τρέχον παράθυρο ή καρτέλα του προγράμματος περιήγησης. Είναι το ανώτατο αντικείμενο στο **BOM** και περιέχει άλλα αντικείμενα που σας λένε σχετικά με το πρόγραμμα περιήγησης.

PROPERTY	DESCRIPTION
<code>window.history</code>	Reference to history object for browser window or tab.
<code>window.innerWidth</code>	Width of window (excluding browser chrome/user interface) (in pixels)
<code>window.pageXOffset</code>	Distance document has been scrolled horizontally (in pixels)
METHOD	DESCRIPTION
<code>window.alert()</code>	Creates dialog box with message (user must click OK button to close it)
<code>window.open()</code>	Opens new browser window with URL specified as parameter (if browser has
<code>window.print()</code>	Tells browser that user wants to print contents of current page

THE WINDOW OBJECT : Παράδειγμα

```
① [var msg = '<h2>browser window</h2><p>width: ' + window.innerWidth + '</p>';  
msg += '<p>height: ' + window.innerHeight + '</p>';  
msg += '<h2>history</h2><p>items: ' + window.history.length + '</p>';  
② [msg += '<h2>screen</h2><p>width: ' + window.screen.width + '</p>';  
msg += '<p>height: ' + window.screen.height + '</p>';  
③ [var el = document.getElementById('info');  
el.innerHTML = msg;  
④ alert('Current page: ' + window.location);
```



browser window

width:1419
height:1100

history

items:1

screen

width:2560
height: 1440

THE DOCUMENT OBJECT MODEL: THE DOCUMENT OBJECT

Το ανώτατο αντικείμενο στο DOM είναι το **DOCUMENT OBJECT**. Αντιπροσωπεύει την ιστοσελίδα που έχει φορτωθεί στο τρέχον παράθυρο ή καρτέλα προγράμματος περιήγησης.

PROPERTY	DESCRIPTION
<code>document.title</code>	Reference to history object for browser window or tab.
<code>document.lastModified</code>	Width of window (excluding browser chrome/user interface) (in pixels)
<code>document.URL</code>	Distance document has been scrolled horizontally (in pixels)
<code>document.domain</code>	Creates dialog box with message (user must click OK button to close it)

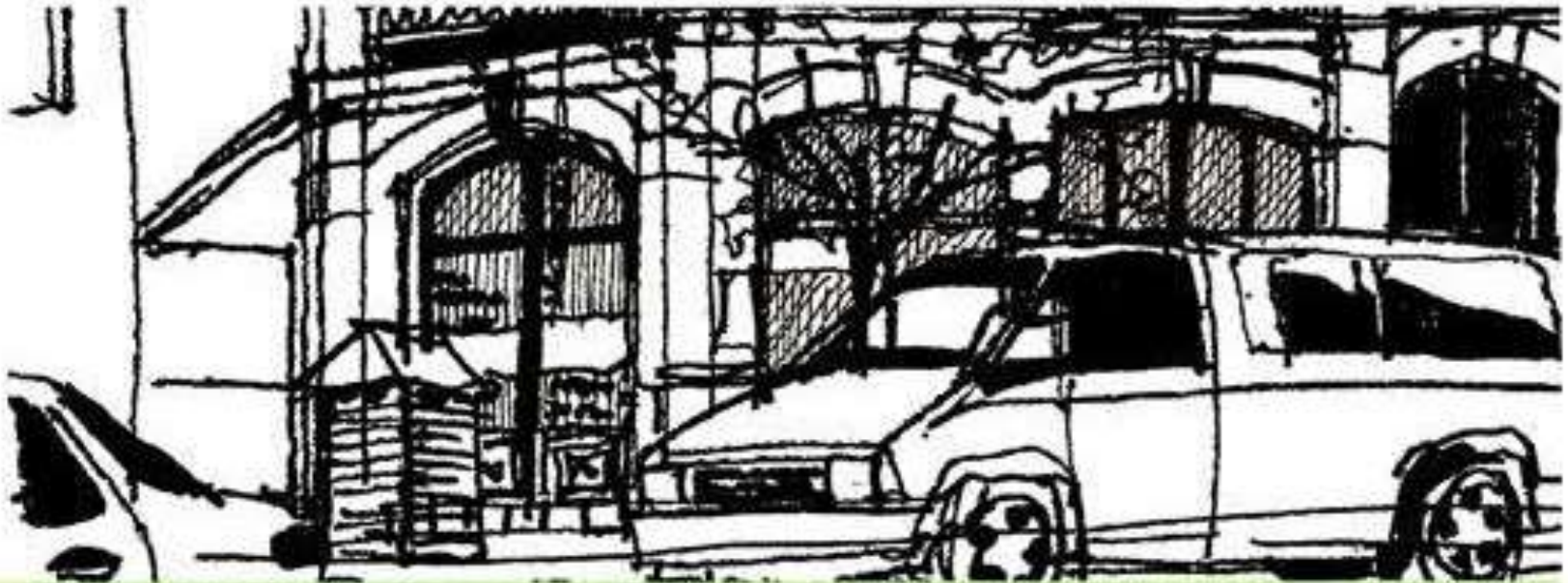
THE DOCUMENT OBJECT MODEL: THE DOCUMENT OBJECT

METHOD	DESCRIPTION
<code>document.write()</code>	Writes text to document
<code>document.getElementById()</code>	Returns element, if there is an element with the value of the id attribute that matches
<code>document.querySelectorAll()</code>	Returns list of elements that match a CSS selector, which is specified as a parameter
<code>document.createElement()</code>	Creates new element
<code>document.createTextNode()</code>	Creates new text node

USING THE DOCUMENT OBJECT

```
① [var msg = '<p><b>page title: </b>' + document.title + '<br />';  
msg += '<b>page address: </b>' + document.URL + '<br />';  
msg += '<b>last modified: </b>' + document.lastModified + '</p>';
```

```
② [var el = document.getElementById('footer');  
el.innerHTML = msg;
```



page title: TravelWorthy

page address: <http://javascriptbook.com/code/co3/document-object.html>

last modified: 03/10/2014 14:46:23

GLOBAL OBJECTS: STRING OBJECT

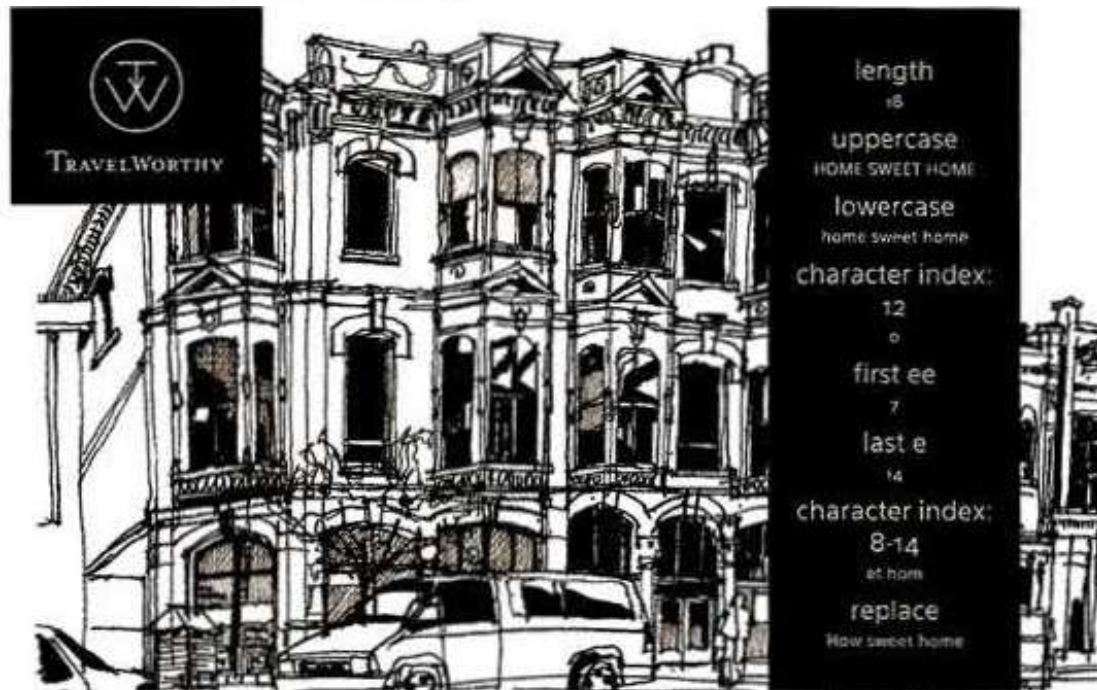
Home sweet home
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

EXAMPLE		RESULT
<code>saying.length;</code>	Home sweet home	16
<code>saying.toUpperCase();</code>	HOME SWEET HOME	'HOME SWEET HOME '
<code>saying.toLowerCase();</code>	home sweet home	'home sweet home '
<code>saying.charAt(12);</code>	Home sweet home	'o'
<code>saying.indexOf('ee');</code>	Home sweet home	7
<code>saying.lastIndexOf('e');</code>	Home sweet home	14
<code>saying.substring(8,14);</code>	Home sweet home	'et hom'
<code>saying.split(' ');</code>	Home sweet home	['Home', 'sweet', 'home', '']
<code>saying.trim();</code>	Home sweet home	'Home sweet home'
<code>saying.replace('me','w');</code>	How sweet home	'How sweet home '

GLOBAL OBJECTS: STRING OBJECT

Home sweet home
0 1 2 3 4 5 6 7 8 9 10 11 12

- ① `var saying = 'Home sweet home ';`
- ② `var msg = '<h2>length</h2><p>' + saying.length + '</p>';`
`msg += '<h2>uppercase</h2><p>' + saying.toUpperCase() + '</p>';`
`msg += '<h2>lowercase</h2><p>' + saying.toLowerCase() + '</p>';`
`msg += '<h2>character index: 12</h2><p>' + saying.charAt(12) + '</p>';`
- ③ `msg += '<h2>first ee</h2><p>' + saying.indexOf('ee') + '</p>';`
`msg += '<h2>last e</h2><p>' + saying.lastIndexOf('e') + '</p>';`
`msg += '<h2>character index: 8-14</h2><p>' + saying.substring(8, 14) + '</p>';`
`msg += '<h2>replace</h2><p>' + saying.replace('me', 'w') + '</p>';`
- ④ `var el = document.getElementById('info');`
`el.innerHTML = msg;`



GLOBAL OBJECTS: NUMBER OBJECT

METHOD	DESCRIPTION
<code>isNaN()</code>	Checks if the value is not a number
<code>toFixed()</code>	Rounds to specified number of decimal places (returns a string)
<code>toPrecision()</code>	Rounds to total number of places (returns a string)
<code>toExponential()</code>	Returns a string representing the number in exponential notation

① `var originalNumber = 10.23456;`

3 decimal places

```
var msg = '<h2>original number</h2><p>' + originalNumber + '</p>';
```

② `msg += '<h2>toFixed()</h2><p>' + originalNumber.toFixed(3); + '</p>';`

③ `msg += '<h2>toPrecision()</h2><p>' + originalNumber.toPrecision(3) + '</p>';`

```
var el = document.getElementById('info');
```

```
el.innerHTML = msg; 3 digits
```

original number

10.23456

3 decimal places

10.235

3 digits

10.2

GLOBAL OBJECTS: MATH OBJECT

PROPERTY	DESCRIPTION
<code>Math.PI</code>	Returns pi (approximately 3.14159265359)

METHOD	DESCRIPTION
<code>Math.round()</code>	Rounds number to the nearest integer
<code>Math.sqrt(<i>n</i>)</code>	Returns square root of positive number, e.g., <code>Math.sqrt(9)</code> returns 3
<code>Math.ceil()</code>	Rounds number up to the nearest integer
<code>Math.floor()</code>	Rounds number down to the nearest integer
<code>Math.random()</code>	Generates a random number between 0 (inclusive) and 1 (not inclusive)

```
var randomNum = Math.floor((Math.random() * 10) + 1);
```

```
var el = document.getElementById('info');  
el.innerHTML = '<h2>random number</h2><p>' + randomNum + '</p>';
```

random number

GLOBAL OBJECTS: DATE OBJECT

METHOD		DESCRIPTION
<code>getDate()</code>	<code>setDate()</code>	Returns / sets the day of the month (1-31)
<code>getDay()</code>		Returns the day of the week (0-6)
<code>getFullYear()</code>	<code>setFullYear()</code>	Returns / sets the year (4 digits)
<code>getHours()</code>	<code>setHours()</code>	Returns / sets the hour (0-23)
<code>getMilliseconds()</code>	<code>setMilliseconds()</code>	Returns / sets the milliseconds (0-999)
<code>getMinutes()</code>	<code>setMinutes()</code>	Returns / sets the minutes (0-59)
<code>getMonth()</code>	<code>setMonth()</code>	Returns / sets the month (0-11)
<code>getSeconds()</code>	<code>setSeconds()</code>	Returns / sets the seconds (0-59)
<code>getTime()</code>	<code>setTime()</code>	Number of milliseconds since January 1, 1970, 00:00:00 UTC (Coordinated Universal Time) and a negative number for any time before
<code>getTimezoneOffset()</code>		Returns time zone offset in mins for locale
<code>toString()</code>		Returns "date" as a human-readable string
<code>toTimeString()</code>		Returns "time" as a human-readable string
<code>toISOString()</code>		Returns a string representing the specified date

CREATING A DATE OBJECT

To specify a date and time, you can use this format:

YYYY, MM, DD, HH, MM, SS
1996, 04, 16, 15, 45, 55

This represents 3:45pm and 55 seconds on April 16, 1996.

The order and syntax for this is:

Year	four digits
Month	0-11 (Jan is 0)
Day	1-31
Hour	0-23
Minutes	0-59
Seconds	0-59
Milliseconds	0-999

Another way to format the date and time is like this:

MMM DD, YYYY HH:MM:SS
Apr 16, 1996 15:45:55

You can omit the time portion if you do not need it.

```
① var today = new Date();
```

```
② var year = today.getFullYear();
```

```
③ [var el = document.getElementById('footer');  
el.innerHTML = '<p>Copyright &copy;' + year + '</p>';
```

