ΠΕΡΙΕΧΟΜΕΝΑ:

ΜΑΘΗΜΑ 1.7: ELEMENT: ΤΟΠΟΘΕΤΗΣΗ

1. Positioning

JS Browser psounis YouTube

Υπενθύμιση από CSS Μάθημα 6.2:

- Αλλάζουμε την τοποθέτηση ενός στοιχείου με το property position.
- Τοποθετείται σχετικά με το «containing element» ή το παράθυρο.
    - Containing Element: Στοιχείο για το οποίο έχει γίνει τοποθέτηση.

Το property offsetParent:

- Επιστρέφει το “containing element” του στοιχείου.
    - Προσοχή ότι αν η τοποθέτηση του στοιχείου είναι fixed, τότε επιστρέφει null

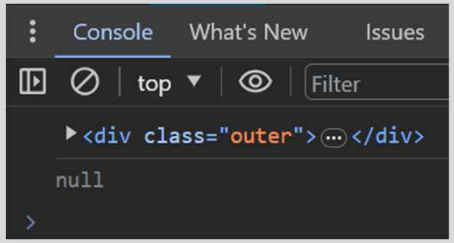
Παράδειγμα 1: element-offsetParent

```html
<div class="outer">
 <div class="inner">
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
 </div>
</div>
```

```css
.outer {
  width: 200px;
  height: 200px;
  padding: 20px;
  border: 1px black solid;
  background-color: lightblue;
  position: relative;
}
.inner {
  border: 1px black solid;
  background-color: lightyellow;
  padding: 20px;
}
```

```css
p {
  border: 1px black solid;
  background-color: lightcoral;
}
p:first-of-type {
  position: relative;
  top: 20px;
  left: 40px;
}
p:last-of-type {
  position: fixed;
  top: 200px;
  right: 50px;
}
```

```js
let p = document.querySelector("p:first-of-type");
console.log(p.offsetParent);
p = document.querySelector("p:last-of-type");
console.log(p.offsetParent)
```

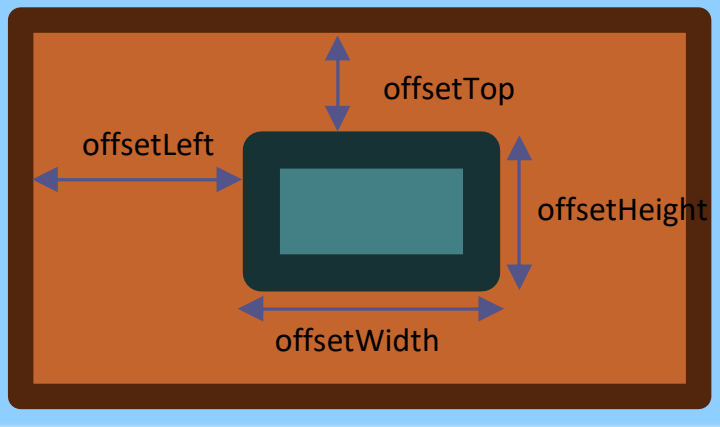


ΜΑΘΗΜΑ 1.7: ELEMENT: ΤΟΠΟΘΕΤΗΣΗ — 1.1. Σε σχέση με το containing element (1/2)

Properties σε σχέση με το containing element:

- Τα παρακάτω properties, επιστρέφουν την απόσταση σε σχέση με το “containing element“:

Property	Επεξήγηση
offsetTop	(κάθετος άξονας) Απόσταση από το εσωτερικό άκρο του border του containing element, ως το εξωτερικό άκρο του border του contained element
offsetLeft	(οριζόντιος άξονας) Απόσταση από το εσωτερικό άκρο του border του containing element, ως το εξωτερικό άκρο του border του contained element
offsetWidth	Πλάτος του containedElement (border+padding+content)
offsetHeight	Ύψος του containedElement (border+padding+content)

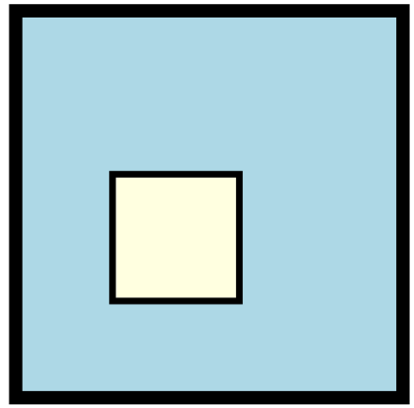


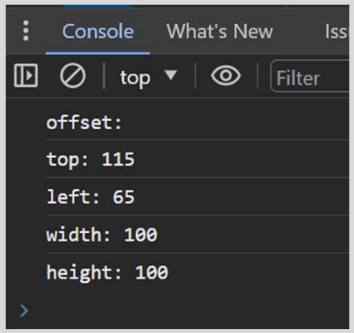
Παράδειγμα 2: element-offset-properties.html

```html
<div class="outer">
  <div class="inner">
  </div>
</div>
```

```css
* {
   box-sizing: border-box;
}
body {
   margin: 30px;
}
.outer {
   width: 300px;
   height: 300px;
   background-color: lightblue;
   position: relative;
   border: 10px black solid;
   padding: 40px;
   margin: 100px;
}
.inner {
   width: 100px;
   height: 100px;
   background-color: lightyellow;
   position: absolute;
   border: 5px black solid;
   padding: 5px;
   margin: 15px;
   left: 50px;
   top: 100px;
}
```

```js
let div =
document.querySelector("div.inner");
console.log("offset:");
console.log("top: " + div.offsetTop);
console.log("left: " + div.offsetLeft);
console.log("width: " + div.offsetWidth);
console.log("height: " + div.offsetHeight);
```





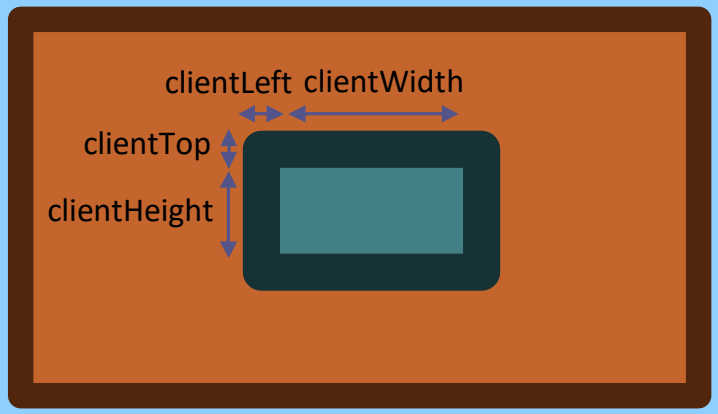
ΜΑΘΗΜΑ 1.7: ELEMENT: ΤΟΠΟΘΕΤΗΣΗ 1.2. Σε σχέση με το containing element (2/2)

JS Browser psounis YouTube

Properties σε σχέση με το containing element:

- Τα παρακάτω properties, επιστρέφουν την απόσταση σε σχέση με το “containing element: “

Property	Επεξήγηση
clientTop	(κάθετος άξονας) Μήκος του top border του στοιχείου
clientLeft	(οριζόντιος άξονας) Μήκος του left border του στοιχείου
clientWidth	Πλάτος του στοιχείου, που περιέχει μόνο το padding (όχι margin, border και scrollbars)
clientHeight	Ύψος του στοιχείου, που περιέχει μόνο το padding (όχι margin, border και scrollbars)



Παράδειγμα 3: element-client-properties.html

HTML

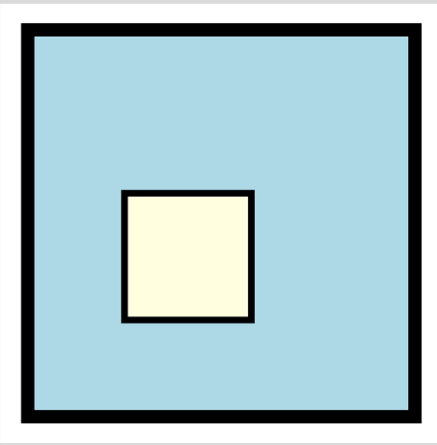
```
<div class="outer">
  <div class="inner">
  </div>
</div>
```

CSS

```
* {
    box-sizing: border-box;
}
body {
    margin: 30px;
}
.outer {
    width: 300px;
    height: 300px;
    background-color: lightblue;
    position: relative;
    border: 10px black solid;
    padding: 40px;
    margin: 100px;
}
.inner {
    width: 100px;
    height: 100px;
    background-color: lightyellow;
    position: absolute;
    border: 5px black solid;
    padding: 5px;
    margin: 15px;
    left: 50px;
    top: 100px;
}
```

JS

```
let div =
document.querySelector("div.inner");
console.log("client:");
console.log("top: " + div.clientTop);
console.log("left: " + div.clientLeft);
console.log("width: " + div.clientWidth);
console.log("height: " + div.clientHeight);
```



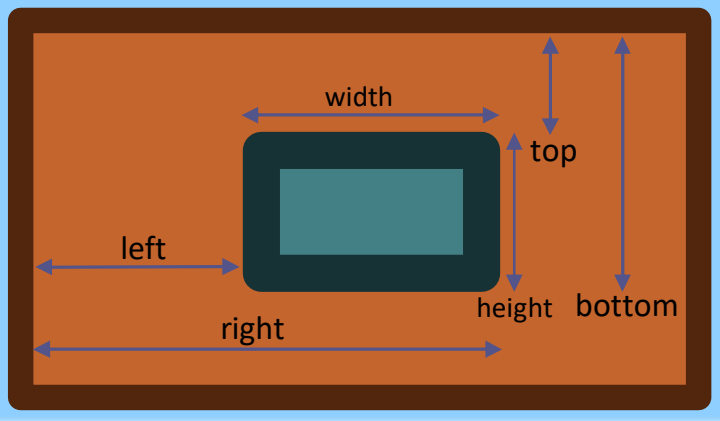
```
▸ border-top-color      rgb(0, 0, 0)
▸ border-top-style      solid
▸ border-top-width      4px
```

```
client:
top: 4
left: 4
width: 92
height: 92
```

Properties σε σχέση με το viewport:

- Με την μέθοδο getBoundingRect() παίρνουμε τη θέση του στοιχείου σε σχέση με το viewport:
- Επιστρεφει ένα αντικείμενο με τα εξής properties:

Property	Επεξήγηση
top	Απόσταση του top του viewport από το top του στοιχείου
right	Απόσταση του left του viewport από το right του στοιχείου
bottom	Απόσταση του top του viewport από το bottom του στοιχείου
left	Απόσταση του left του viewport από το left του στοιχείου
width	Πλάτος του στοιχείου
height	Ύψος του στοιχείου



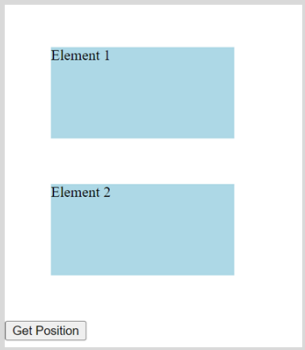
Παράδειγμα 4: getBoundingRect.html

```html
<div class="box" id="box1">Element 1</div>
<div class="box" id="box2">Element 2</div>

<button id="getPositionBtn">Get Position</button>
```

```js
document.getElementById('getPositionBtn').addEventListener('click', function() {
  const box1 = document.getElementById('box1');
  const rect = box1.getBoundingClientRect();

  console.log('Top:', rect.top);
  console.log('Right:', rect.right);
  console.log('Bottom:', rect.bottom);
  console.log('Left:', rect.left);
  console.log('Width:', rect.width);
  console.log('Height:', rect.height);
});
```




```
Top: 76
Right: 258
Bottom: 176
Left: 58
Width: 200
Height: 100
```

2.1. Παράδειγμα: Εμφάνιση Διαλόγου

Εμφάνιση Διαλόγου ώστε να χωράει στο παράθυρο

- Με χρήση των διαστάσεων μπορούμε να εμφανίζουμε modals (παράθυρα διαλόγου) ώστε να χωράνε στη σελίδα

Σημείωση:

- Στο παράδειγμα χρησιμοποιείται το window.innerHeight που είναι το ύψος του viewport (βλ. ενότητα 3)

Παράδειγμα 5: example-modal-positioning.html

```html
<button class="button" id="openModalBtn">Open Modal</button>

<div class="modal hide" id="modal">This is a modal dialog.</div>
```

```css
.button {
  margin: 100px;
  padding: 10px 20px;
  background-color: blue;
  color: white;
  border: none;
  cursor: pointer;
}
```

```css
.modal {
  position: absolute;
  width: 200px;
  padding: 20px;
  background-color: lightgray;
  border: 1px solid black;
}

.hide {
  display: none;
}
```

```js
const openModalBtn = document.getElementById('openModalBtn');
const modal = document.getElementById('modal');

openModalBtn.addEventListener('click', () => {
  // Temporarily show the modal for accurate measurement
  modal.classList.remove('hide');
  const modalHeight = modal.offsetHeight;
  modal.classList.add('hide');

  // Get the button's position and dimensions
  const buttonRect = openModalBtn.getBoundingClientRect();
  // Get viewport height to check if there is enough space below the button
  const viewportHeight = window.innerHeight;
  // Calculate the modal's top position
  let topPosition = buttonRect.bottom;

  // If there isn't enough space below, place the modal above the button
  console.log(buttonRect.bottom, modalHeight, viewportHeight);
  if (buttonRect.bottom + modal.offsetHeight > viewportHeight) {
    topPosition = buttonRect.top - modal.offsetHeight;
  }
  // Set the modal's position
  modal.style.top = `${topPosition}px`;
  modal.style.left = `${buttonRect.left}px`;
  // Display the modal
  modal.style.display = 'block';
});
```