

# Informatietechnologie 2

## JavaScript - het window object + extras

**Kristof Michiels**

# Het Window Object

# Het Window object

- Vertegenwoordigt een venster dat een DOM document bevat
- Document is onderdeel van dit object, ze zijn vaak onderling uitwisselbaar bruikbaar
- De document-eigenschap verwijst naar het DOM document dat in de venster wordt ingeladen
- Het Window object bevat heel wat functionaliteit, met name zaken die globaal beschikbaar moeten zijn

```
window.document.querySelector("p").innerHTML = "hallo";
```

# Events: keydown en keyup

- We werkten tot nog toe vooral met events als click en mouseover
- Met keydown en keyup kunnen we toetsindrukken opvangen

```
window.addEventListener("keydown", function(event) {  
    if (event.key === "v") {  
        document.body.style.background = "violet";  
    }  
});  
window.addEventListener("keyup", function(event) {  
    if (event.key === "v") {  
        document.body.style.background = "";  
    }  
});
```

# Event: mousemove

- Gaat af als de gebruiker met de muispijl over het browservenster beweegt
- De plek waar de muispijl zich bevindt vind je via `pageX` en `pageY`

```
let mijnParagraaf = document.querySelector("p");
window.addEventListener("mousemove", function(event) {
    mijnParagraaf.innerHTML = "Muiscoördinaten: X="
    + event.pageX + "px, Y=" + event.pageY + "px";
});
```

# Scroll-event en pageXOffset/pageYOffset eigenschap

- Gaat af als de gebruiker verticaal en horizontaal scrollt
- window.pageXOffset en window.pageYOffset geven aan hoeveel reeds werd gescrolld

```
let mijnParagraaf = document.querySelector("p");
let mijnH1 = document.querySelector("h1");
window.addEventListener("scroll", function() {
    mijnParagraaf.innerHTML = "pageXOffset: " + window.pageXOffset +
        "px, pageYOffset: " + window.pageYOffset + "px";
    if (window.pageYOffset > 150) {
        mijnH1.classList.add("onzichtbaar");
    } else {
        mijnH1.classList.remove("onzichtbaar");
    }
});
```

# window.scroll

- met window.scroll kan je de browser een nieuwe scrollpositie doen innemen

```
let mijnKnoppen = document.querySelectorAll("a");
let i = 0;
let scrollAantal = 0;
while (i < mijnKnoppen.length) {
    mijnKnoppen[i].addEventListener("click", function(event) {
        window.scroll({
            top: (window.innerHeight*this.getAttribute("data-factor")),
            left: 0,
            behavior: 'smooth'
        });
        event.preventDefault();
    }, false);
    i++;
}
```

# Resize-event en innerWidth/innerHeight eigenschap

- Resize gaat af als de gebruiker het venster van omvang doet veranderen
- `window.innerWidth` en `window.innerHeight` geven je (los van het resize-event) de breedte en hoogte van de pagina, in pixel

```
let mijnParagraaf = document.querySelector("p");
mijnParagraaf.innerHTML = "Schermafmetingen: X="
  + window.innerWidth + "px, Y=" + window.innerHeight + "px";

window.addEventListener("resize",function() {
  mijnParagraaf.innerHTML = "Schermafmetingen: X=" + window.innerWidth + "px,
    Y=" + window.innerHeight + "px";
});
```



# Enkele element eigenschappen en methods

# offsetWidth en offsetHeight-eigenschappen

Geven ons de breedte en hoogte die een element inneemt in pixels, met inbegrip van de eventuele border

```
let mijnDiv = document.querySelector("div");
oWidth = mijnDiv.offsetWidth;
oHeight = mijnDiv.offsetHeight;
let mijnPara = document.querySelector("p");
mijnPara.innerHTML = "OffSetWidth: " + oWidth + "px, OffSetHeight: "
+ oHeight + "px (dit is de totale ruimte dat een element in neemt)";
```

# clientWidth en clientHeight-eigenschappen

Geven ons de ruimte die een element inneemt in pixels, ZONDER de border

```
let mijnDiv = document.querySelector("div");
cWidth = mijnDiv.clientWidth;
cHeight = mijnDiv.clientHeight;
let mijnPara = document.querySelector("p");
mijnPara.innerHTML = "clientWidth: " + cWidth + "px, clientHeight: " +
cHeight + "px (dit is de ruimte dat een element in neemt zonder border)";
```

# de getBoundingClientRect-method

- Geeft een object terug met verschillende eigenschappen zoals top, bottom, left en right
- Deze bevatten de positie uitgedrukt in px relatief ten opzichte van de linkerbovenhoek van het scherm

```
let mijnDiv = document.querySelector("div");
let mijnP = document.querySelector("p");
let getBottom = mijnDiv.getBoundingClientRect().bottom;
let getHeight = mijnDiv.getBoundingClientRect().height;
let getLeft = mijnDiv.getBoundingClientRect().left;
let getRight = mijnDiv.getBoundingClientRect().right;
let getTop = mijnDiv.getBoundingClientRect().top;
let getWidth = mijnDiv.getBoundingClientRect().width;
mijnP.innerHTML += "<br>bottom: " + getBottom + "px<br>height: " + getHeight
+ "px" + "<br>left: " + getLeft + "px" + "<br>right: " + getRight + "px" +
"<br>top: " + getTop + "px" + "<br>width: " + getWidth + "px";
```

# Positioning en animatie

# Positioning en animatie

- We kunnen positioning gebruiken om een animatie te creëren
- Je doet dit door vanuit JavaScript de left, top, right en bottom waarden te veranderen

# Positioning en animatie

```
let mijnDiv = document.querySelector("div");
let maxToLeft = window.innerWidth - 150;
window.addEventListener("scroll", function() {
  mijnDiv.style.bottom = "-" + (window.pageYOffset/10) + "rem";
  if (window.pageYOffset > maxToLeft) {
    mijnDiv.style.left = 0;
    mijnDiv.style.backgroundColor = "red";
  } else {
    mijnDiv.style.left = "";
    mijnDiv.style.backgroundColor = "";
    mijnDiv.style.right = (window.pageYOffset/10) + "rem";
  }
});
```

## IT2 - JavaScript - het window object + extras

[kristof.michiels01@ap.be](mailto:kristof.michiels01@ap.be)