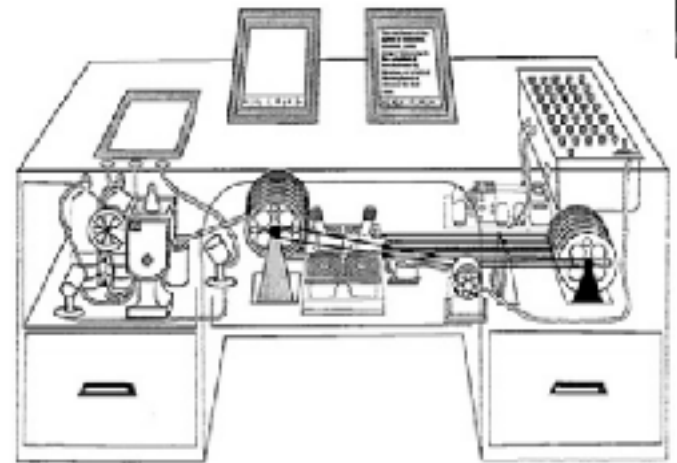


Développement d'interface utilisateur pour le Web

Intro au développement Web

- À l'origine, le WWW a fourni l'accès à des documents du réseau du CERN
- Le lien hypertexte permet d'ouvrir rapidement d'autres pages Web, envisagées par Vannevar Bush



Diviser et conquérir

Une page Web repose sur trois composantes:

Contenu → HTML

Texte, images, animations, vidéos, etc.

Présentation → CSS

Comment il apparaîtra à travers un navigateur Web

Comportement → JavaScript

Interaction en temps réel (validation, tri, ...)

Documents HTML

- Les pages Web sont créées à l'aide du langage HTML (Hypertext Markup Language)
- Un langage de balisage (markup language), c.à.d. un ensemble de caractères ou de symboles qui définissent la structure logique d'un document

Syntaxe HTML de base

- HTML est un format de texte reposant sur des balises (tags)
- Les tags sont entre crochets (<>) et se composent d'une tag d'ouverture et d'une tag de fermeture
- Tags HTML
 - Déclarer des éléments (ex. image, canvas, svg, video, sound, button, checkbox, menu, textfield, etc...)
 - Décrire le contenu (ex. si le texte doit être un titre (h1), un paragraphe (p), accentué (em), une citation, etc ...)
 - Structurer le contenu
- Tutoriel pour l'apprentissage du HTML

<http://www.htmldog.com/>

<http://www.sitepoint.com/html/>

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

CSS

Cascading Style Sheets

- Un seul morceau d'informations de formatage CSS (par exemple l'alignement du texte) est appelé style
- Cascading désigne la possibilité pour les pages Web d'utiliser des informations CSS provenant de plus d'une source

CSS

Couches d'une page Web:

- Contenu
Texte, images, animations, vidéos, etc.
- Présentation
Comment il apparaîtra à travers un navigateur Web
- Comportement
Interaction en temps réel (validation, tri, d & d)

CSS sépare la présentation du contenu

CSS

- Propriétés CSS:
 - Les styles CSS ont deux parties séparées par deux points (:)
 - La propriété fait référence à un style CSS spécifique
 - La valeur qui lui est attribuée détermine les caractéristiques visuelles du style
 - `color:red`
- Ensemble, une propriété CSS et sa valeur sont une *déclaration* ou une déclaration de style

Lien entre CSS et HTML

- Feuilles de style *inline* et *internal* incorporent des déclarations css dans des fichiers html
- Feuilles de style externes, un document de texte distinct contenant des déclarations de style utilisées par plusieurs documents HTML

mywebpage.html

```
<head>
```

```
  < link rel="stylesheet" type="text/css" href="mycss.css" />
```

```
</head>
```

mycss.css

```
h1{font-family:Arial}  //all h1 tags
```

Syntaxe générale

- Un sélecteur (où appliquer les caractéristiques visuelles)
- Et paires de propriété: valeur

```
body {font-family:Arial ; font-size:9pt}
```

- Insensible aux majuscules,
- Les espaces blancs et les sauts de ligne ignorés
- Les sélecteurs peuvent être complexes (unions, intersections, etc.)
- Commentaires: `/* this is a comment */`

e.g. <http://developer.mozilla.org>
<http://css-tricks.com/>

Pourquoi CSS?

- Facile à maintenir
 - Changement une fois appliqué partout
- Mise en cache CSS = moins de bande passante + chargement rapide
- Flexible
 - Peut charger différents CSS dans différentes situations
 - par exemple p.r.à. l'appareil (ex PC, mobile, montre, ... plus tard)

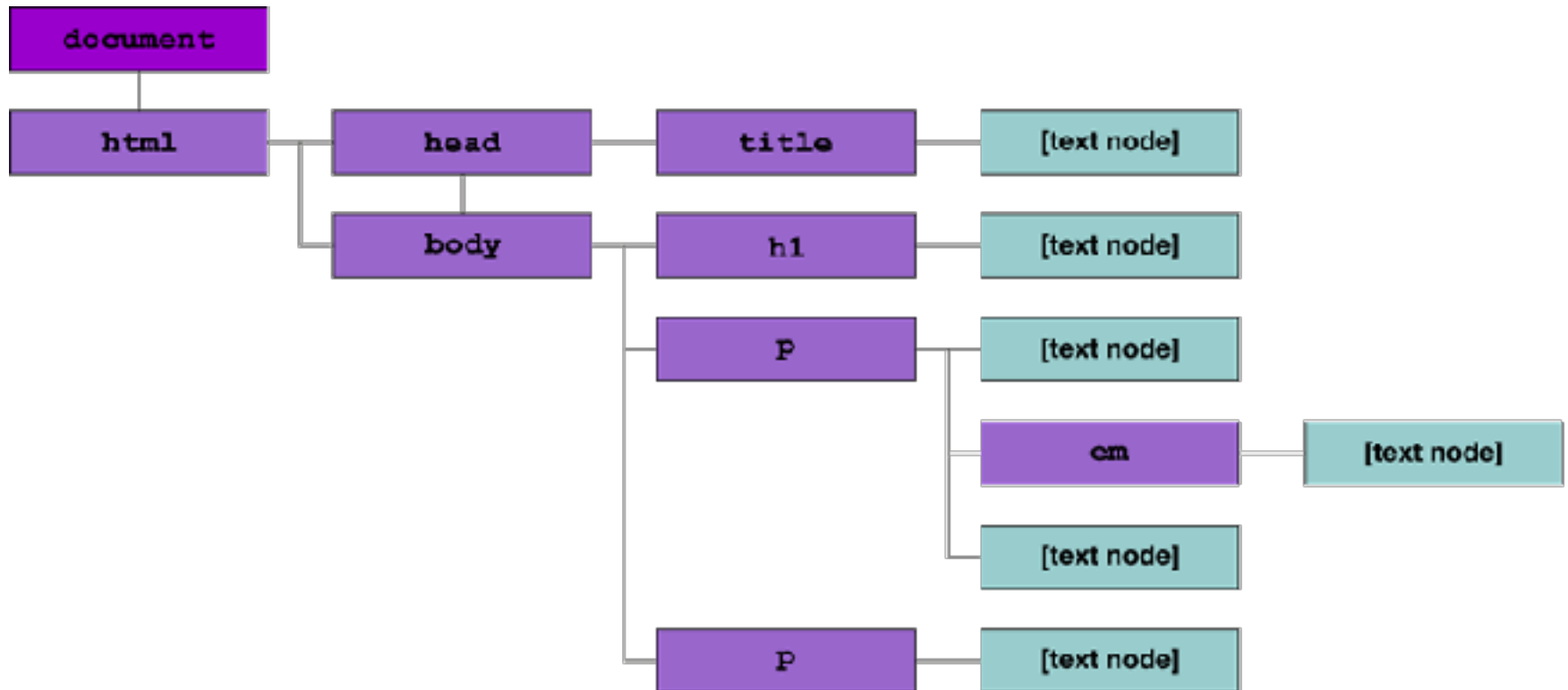
CSS mise en page

block vs inline

- éléments HTML block-level
 - Contiennent des blocs (box) et commencent sur de nouvelles lignes
 - ex. `<h1>...<h6>, <p>, , , , <table>, <tr>, <td>, <div> ...`
- éléments HTML inline (text) level
 - Doivent être imbriqués dans des blocs, peuvent contenir du texte ou d'autres éléments en ligne, ne commencent pas sur de nouvelles lignes
 - ex. `, , <a>, , <abbr>, ...`
- CSS aide à définir leurs propriétés visuelles

DOM

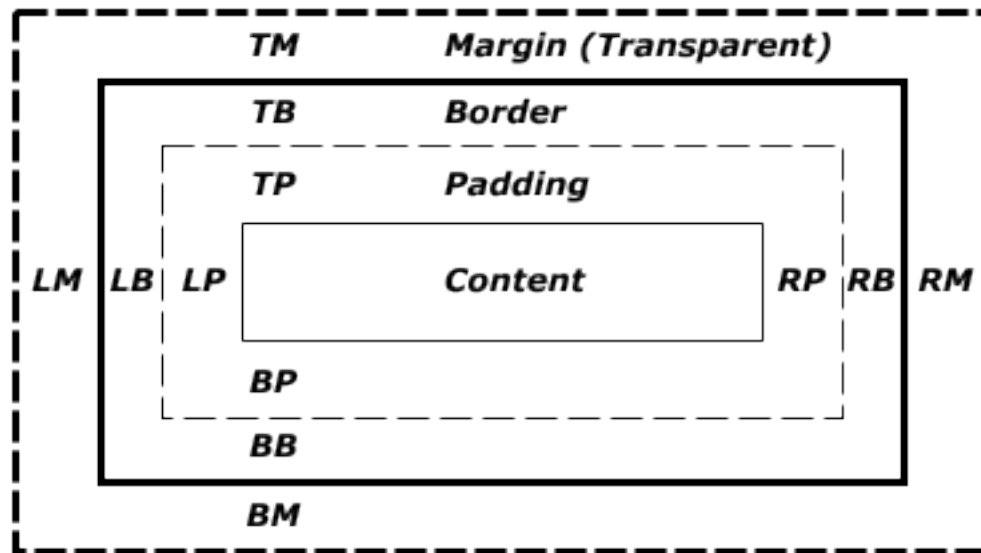
- Le navigateur construit un modèle d'objet de document (document object model ou DOM), c'est-à-dire un arbre de noeuds



DOM

- Le navigateur construit un DOM
- Chaque nœud est rendu sous la forme de 0 ou plusieurs zones:
 - Les éléments inline génèrent des boîtes en ligne
 - Les elements groupent des boîtes
 - En utilisant css on modifie leurs propriétés visuelles
 - Peut fixer la taille d'une boîte (`width`, `height`)
 - ...

Propriétés: le modèle boîte (box model)



- Margin edge
- Border edge
- - - Padding edge
- Content edge

Propriétés: the box model



css3 menus

css3 animations

css3 transforms

CSS et menus faciles

- La navigation facile est importante
- Barre de navigation = une (jolie) liste de liens

```
<nav>
<ul>
  <li><a href="default.asp">Home</a>
    <ul>
      <li><a href="about.asp">About</a></li>
    </ul></li>
  <li><a href="news.asp">News</a></li>
  <li><a href="contact.asp">Contact</a></li>
</ul>
</nav>
```

CSS et menus faciles

- dans css supprimer les (par défaut) bullets et rembourrage (padding)

```
nav ul {  
    list-style-type:none;  
    margin:0;  
    padding:0;  
}
```

- masquer les sous-menus et le menu déroulant

```
nav ul ul { display: none; }  
nav ul li:under > ul { display: block;}
```

- Autres sélecteurs "événement": link, visited, hover, active, focus, selection, checked, etc.

CSS et menus faciles

- Une barre verticale

```
nav a {  
    display: block;  
    width: 60px;  
}
```



- ne barre horizontale

```
nav li { float: left; }  
nav a {  
    display: block;  
    width: 60px;  
}
```



CSS3 animations

- Les animations sont des transitions entre les configurations de style
 - Style décrivant l'animation CSS
 - Images clés (keyframes) pour les états de début et de fin de style (et éventuels points intermédiaires)

```
h1 {  
    animation-duration: 3s;  
    animation-name: slidein;  
    animation-iteration-count: infinite;  
}  
  
@keyframes slidein {  
    from { margin-left: 100%; width: 300%;}  
    to   { margin-left: 0%; width: 100%; }  
}
```

plus à https://developer.mozilla.org/en-US/docs/CSS/Using_CSS_animations

CSS3 animations

- Vous pouvez ajouter des écouteurs aux événements d'animation (listeners), tels que
 - Début d'animation, fin d'animation et début d'une nouvelle itération
 - Ils incluent des informations sur le moment où l'événement a eu lieu et le nom de l'animation qui l'a déclenchée

CSS3 transforms

- Avec la transformation, nous modifions l'espace des coordonnées des éléments (translated, rotated, scaled, et skewed)
- Empiler le contexte (appliqué l'un après l'autre)

```
transform: none
```

```
transform: matrix(1.0, 2.0, 3.0, 4.0, 5.0, 6.0)
```

```
transform: translate(12px, 50%) // translateX, translateY
```

```
transform: scaleX(2) // scale, scaleY
```

```
transform: rotate(0.5turn) // degrees/rad rotateX ...
```

```
transform: skewX(30deg) // skewY
```

```
transform-origin ....
```

```
h1 {-webkit-transform: rotate(45deg)}
```

- (si vous avez eu un cours en infographie)

- <https://developer.mozilla.org/en-US/docs/CSS/transform>



javascript

Qu'est-ce que JavaScript (JS)?

- Langage dynamique et fonctionnel (comme java, C #)
 - Syntaxe influencée par C
 - Noms et conventions de nommage de Java, O-O
 - Entrée traitée avec les auditeurs
- Peut être interprété par les navigateurs Web
- Peut être utilisé pour la programmation client Web
- Peut être utilisé pour la programmation du serveur (par exemple Node.js)

JavaScript (JS)

- Importance pour nous
 - Utilisé pour fournir l'interactivité aux sites Web et aux applications Web
- Nous permet de changer
 - La liste des feuilles de style (CSS) du document
 - Les règles d'une feuille de style (CSS)
 - Les différents éléments du DOM, indépendamment de la feuille de style utilisée

JavaScript (JS)

- Structures de programmation classiques
 - Déclarations, fonctions, commentaires, `IF ... THEN, FOR, WHILE, ...`
 - événements (onmouseover, onclick, onkeyup, etc)
 - accès au DOM du html
- Exemples et tutoriels
<https://developer.mozilla.org/>

JS scripts

```
<!DOCTYPE html>  
<html>  
<body>
```

```
<p>Changing web page colors with different delays.</p>
```

```
<script>  
// JS script will be called as the page loads  
setTimeout("document.bgColor='white'", 1000)  
setTimeout("document.bgColor='lightpink'", 1500)  
setTimeout("document.bgColor = 'pink'", 2000)  
setTimeout("document.bgColor = 'deeppink'", 2500)  
setTimeout("document.bgColor = 'red'", 3000)  
setTimeout("document.bgColor = 'tomato'", 3500)  
setTimeout("document.bgColor = 'darkred'", 4000)  
</script>
```

```
</body>  
</html>
```

Changing web page colors with different delays.

JS manipulation du Canvas

- `<canvas>` est un élément HTML5, utilisé pour dessiner des graphiques à l'aide de scripts (par exemple JS).

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
</canvas>
```

```
<script>
```

```
// JS script will be called when page loads, look for ID myCanvas, and draw a line
```

```
var c = document.getElementById("myCanvas");
```

```
var ctx = c.getContext("2d");
```

```
ctx.moveTo(0,0);
```

```
ctx.lineTo(200,100);
```

```
ctx.stroke();
```

```
</script>
```

```
</body>
```

```
</html>
```



JS mettre en œuvre des rappels (callbacks/listeners)

```
<!DOCTYPE html>
<html>
<body>
```

<p>Click the button to loop from 1 to 6, to make HTML headings.</p>

```
<button onclick="myFunction()">Try it</button>  <!-- adding a callback function to widget button -->
<div id="demo"></div>
```

```
<script>
```

// JS script provides the implementation of the callback function (equivalent to listener functions)

```
function myFunction()
{
  var x="",i;
  for (i=1; i<=6; i++)
  {
    x=x + "<h" + i + ">Heading " + i + "</h" + i + ">";
  }
  document.getElementById("demo").innerHTML=x;
}
</script>
```

```
</body>
</html>
```

Click the button to loop from 1 to 6, to make HTML headings.

[Try it](#)

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Bibliothèque JQuery pour JS

- JQuery est une bibliothèque pour JS
- Il fournit une API inter-navigateur pour faciliter
 - Manipulation HTML / DOM
 - Gestion des événements DOM
 - Manipulation CSS
 - Effets et animations
 - AJAX (communication client-serveur)
 - Autres utils

JQuery Syntaxe

- sélectionner des éléments HTML et les manipuler
- Syntaxe: **\$(selector).action()** (au lieu de getElementById)
 - Le \$ définit / accède jQuery
 - Le (selector) trouve les éléments HTML
 - L'action() jQuery est effectuée sur l'élément (s)
- Exemples:
 - `$(this).hide()` - masque l'élément courant.
 - `$("p").hide()` - masque tous les éléments `<p>`.

Bibliothèque JQuery pour JS

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
```

```
<script>
```

```
$(document).ready( function(){  
    $("button").click( function(){  
        $("p").hide();  
    });  
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h2>This is a heading</h2>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
<button>Click me</button>
```

```
</body>
```

```
</html>
```

This is a heading

This is a paragraph.

This is another paragraph.

Click me

This is a heading

Click me

Cooler stuff in the TA !!

plus sur JQuery

- Les fonctions peuvent avoir des "callbacks" pour ordonner des événements (fonctions équivalentes à des auditeurs/listeners)

```
$( "button" ).click(function(){  
    $( "p" ).hide("slow",function(){  
        alert("The paragraph is now hidden");  
    });  
});
```

- et peut être enchaîné

```
$( "#p1" ).css("color","red").slideUp(2000).slideDown(2000);
```

plus sur JQuery

- Beaucoup d'extensions JQuery pour
 - création et manipulation de widgets
 - dessin de widgets
 - des extensions d'interaction,
par ex. <http://jqueryui.com/>

plus sur JQuery

JQuery fournit

Interactions (drag, drop, resize, select, sort)

Widgets (button, list, menus, sliders, etc)

Effets et animations (show/hide, color changes)

<http://jqueryui.com/demos/>

Comment dessiner / interagir

Il existe deux façons de dessiner sur une page Web

- Utilisez le `<canvas>`, un élément HTML5 pour le dessin (Comme nous l'avons vu avec JS de base)
- Utilisez `<svg>` (Scalable Vector Graphics), un langage XML pour la création de graphiques

Pour les deux, l'interaction se fait par script (JS et / ou JQuery)

En classe, nous utiliserons une variante de HTML Canvas

Comment dessiner / interagir

- <Canvas> est un élément HTML5
 - Utilisé pour dessiner des graphiques à l'aide de scripts (par exemple JS / JQuery)
 - Bon pour les graphiques, les compositions de photos ou les animations


```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="canvas" width="300" height="300"></canvas>
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
```

```
<script>
```

```
//get a reference to the canvas
```

```
var ctx = $('#canvas')[0].getContext("2d");
```

```
var x = y = 50;
```

```
//draw a rect at x, y
```

```
ctx.rect(x,y,50,50);
```

```
ctx.fillStyle = 'black';
```

```
ctx.fill();
```

```
$(document).mousemove(onMouseMove);
```

```
// attach an callback function to entire web page
```

```
function onMouseMove(evt) {
```

```
  if (evt.pageX > x && evt.pageX < x+50 && evt.pageY > y && evt.pageY < y+50) {
```

```
    ctx.fillStyle = 'red'; }
```

```
  else{
```

```
    ctx.fillStyle = 'black'; }
```

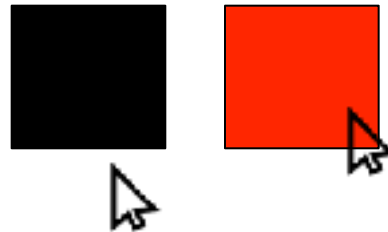
```
  ctx.fill();
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```



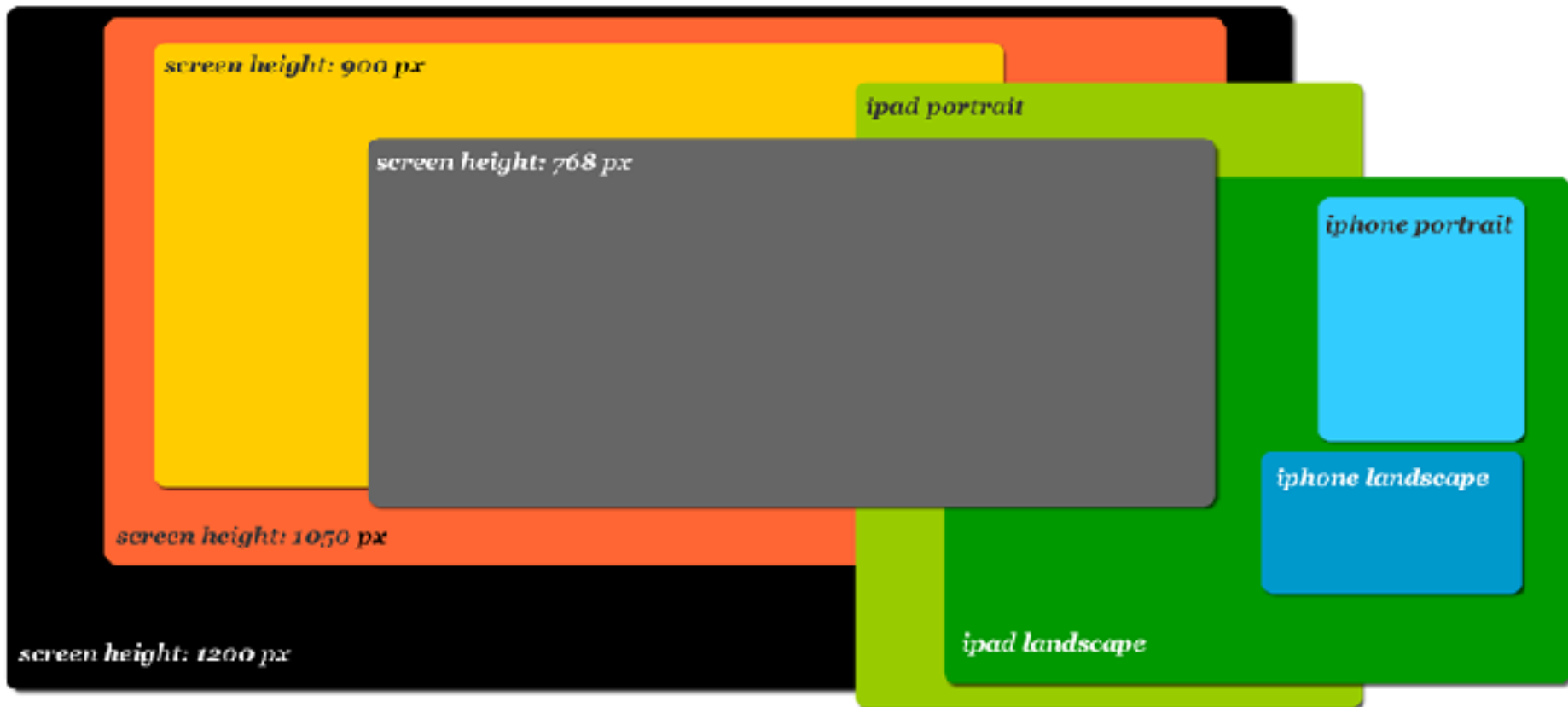
... media queries ...
(responsive web design)

Pourquoi CSS?

- Flexible (peut charger différents CSS dans différentes situations, par exemple des périphériques)
- Comment savons-nous quel appareil / résolution?
- Mais aussi, comment concevons-le (plus tard ...)

resolution examples

quelques resolutions pour iphones (<http://www.websitedimensions.com/>)



sites pour tester votre page:

e.g. <http://quirktools.com/screenfly/>, <http://www.viewlike.us/index.php>

media queries

- Dès CSS2 **media types** (device)

`screen, braille, speech, ...`

- Dans CSS3 **media queries** (device capabilities)

`width & height of browser window,`

`device-width, device-height or device-aspect-ratio`

`orientation (landscape or portrait in phone)`

`resolution (dpi)`

media queries

- Nous interrogeons le type de média et capacités

`@media screen and (min-device-width:481px) and ..`

- et ...
 - Nous créons des blocs de style pour cette requête
 - ... ou appelons une feuille de style différente

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

<http://css-tricks.com/resolution-specific-stylesheets/>

media queries

ex. sur votre fichier CSS des pour de media différentes

```
/* all screens */
```

```
#mypar{ font-size:12px;background-color:#9F0;}
```

```
/* large screen (1440px or more) */
```

```
@media screen and {min-width:1440px} {
```

```
    #mypar{ font-size:18px;background-color:#F90;}
```

```
}
```

```
...
```

media queries

ex. ppeler un autre fichier CSS

```
< link type = "text/css" href="my_default_css.css" />
```

puis l'annuler dans des conditions spécifiques

```
< link rel="stylesheet" type="text/css" media="only screen  
and (max-device-width:480px)" href="small-device.css" />
```

...

media queries

ISO



from mediaqueri.es