

# Intégration web FTW

# Intégrer, c'est quoi?

- Transposer les maquettes graphiques en pages web, en assemblant tous les éléments (textes, images, sons, etc.) dans le respect des normes et des standards.
- Métiers liés: développeur web, architecte web, (webmaster)

**Rappels...**

# Les débuts du WWW

- 1989 (~25ans)
- Créé par Tim Berners-Lee
- Première page web
- Internet existe déjà depuis 60ans! (fin des années 50)

# Le WWW en 1989

- Repose sur:
  - Nom de domaine (DNS)
  - URLs (protocole HTTP)
  - Langage HTML

# Le WWW, 25 ans après

- Repose sur:
  - Nom de domaine (DNS)
  - URLs (protocole HTTP)
  - Langage HTML
- Et quelques fioritures:
  - Images (1993)
  - JavaScript (1995)
  - CSS (1996)
  - etc.

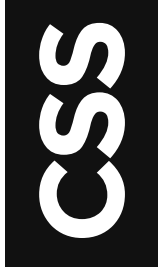
# Techniquement...

- Nom de domaine (DNS: Domain Name System)
- URLs (protocole HTTP)
- Langages client
  - HTML, CSS, JavaScript
- Langages Serveur
  - PHP, JAVA, .Net, JavaScript (Node.js)

# HTML

- HyperText Markup Language
- Langage à balises de données hypertextes
- Définit:
  - le contenu de la page (textes, images, vidéos, etc)
  - la sémantique de ces éléments
  - la structure
  - les liens entre les ressources
- Doit permettre l'accessibilité des données





- Cascading Style Sheets
- Feuilles de styles en cascade
- Définit:
  - le rendu visuel des documents XML
  - un peu d'interactivité visuelle

# JavaScript

- Langage de programmation de script
- A ne confondre avec JAVA
- Utilisé pour
  - interactivité des pages
  - interagir avec le DOM
  - logique applicative

# 3 langages = 3 pôles

Séparation des tâches

- Contenu
- Mise en forme
- Interactivité

**Hands-on!**

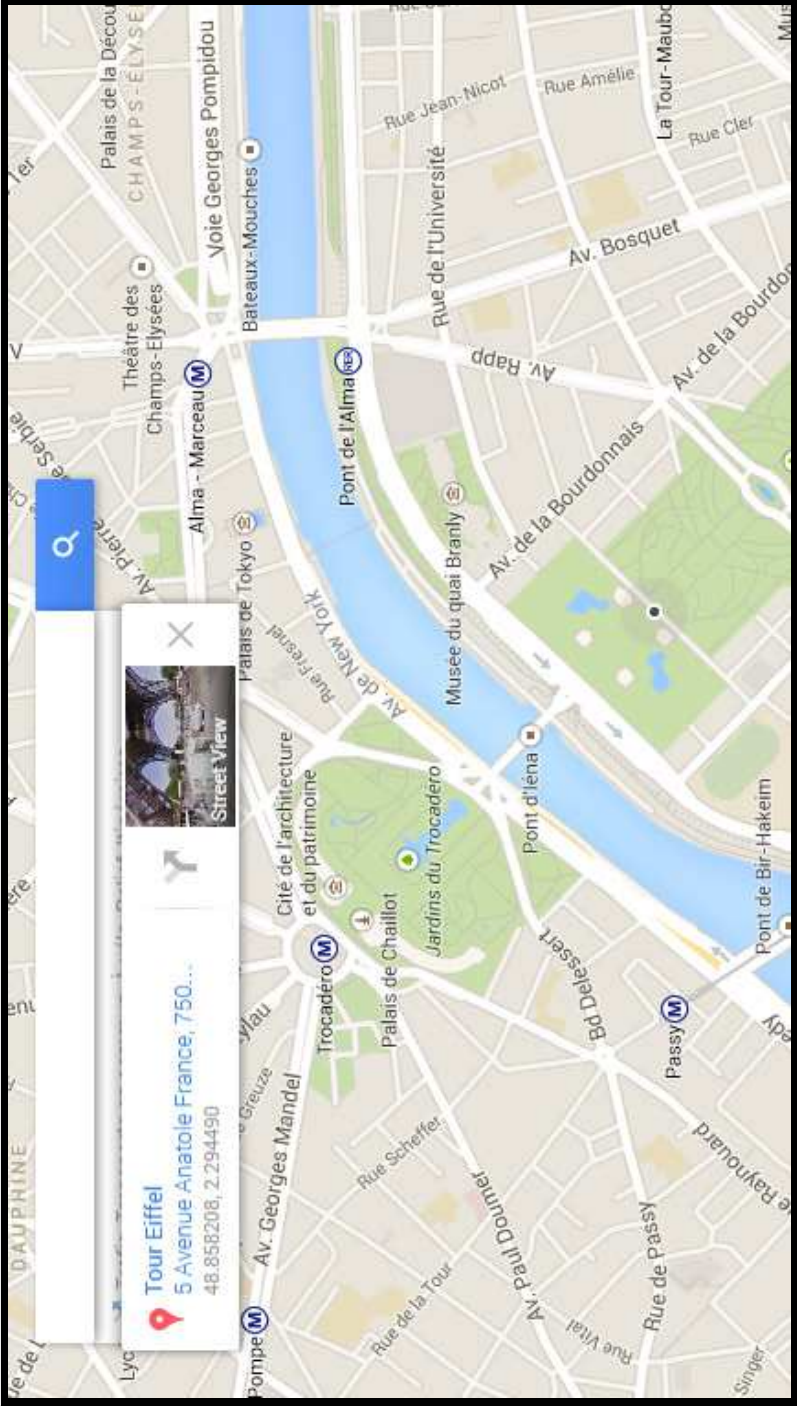
# Aucune différence de techno

- Site web vitrine
- Site e-commerce
- Site applicatif
- Mais également:
  - Certaines applications mobiles
  - Certaines applications natives
- Rien a changé techniquement, et pourtant on est loin des débuts!

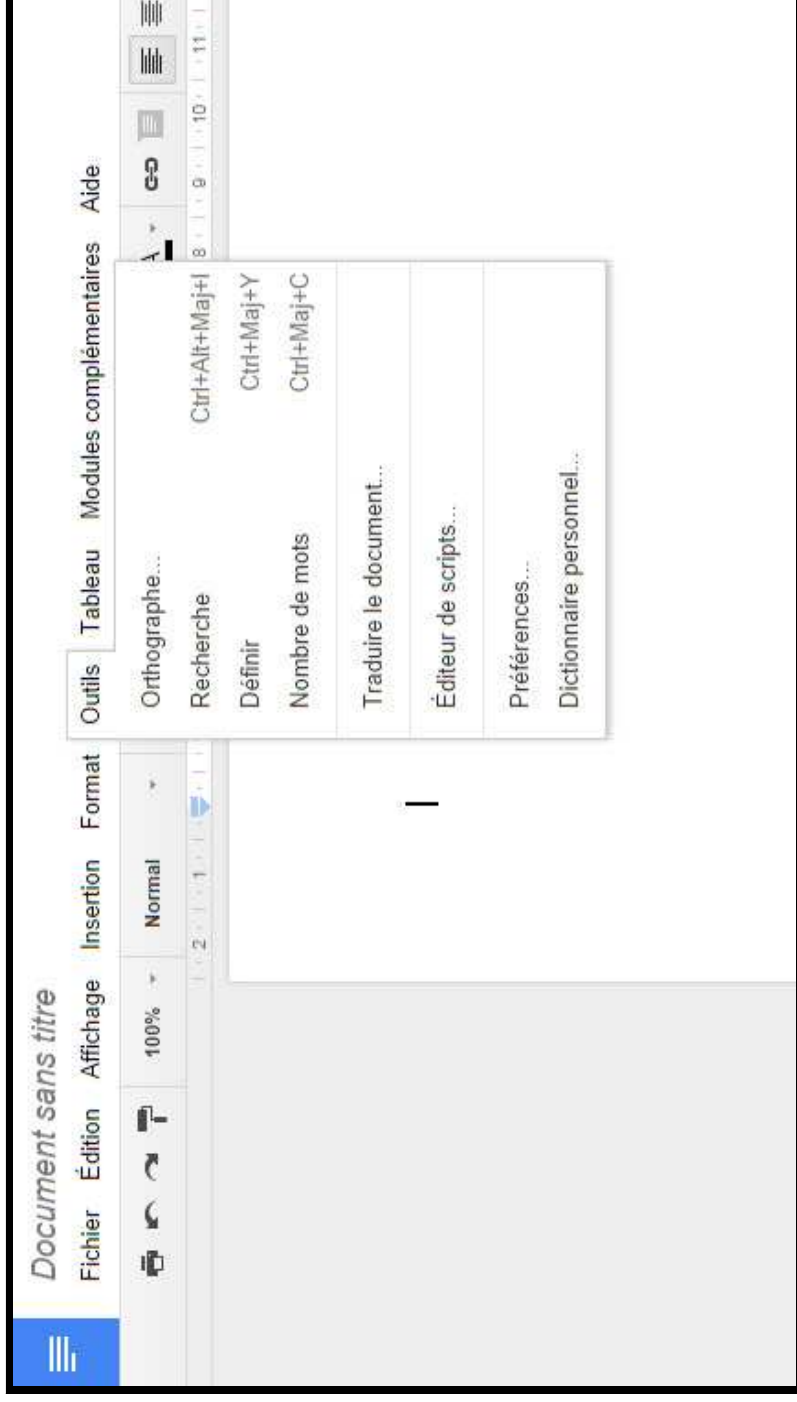
# Facebook



# Google Maps

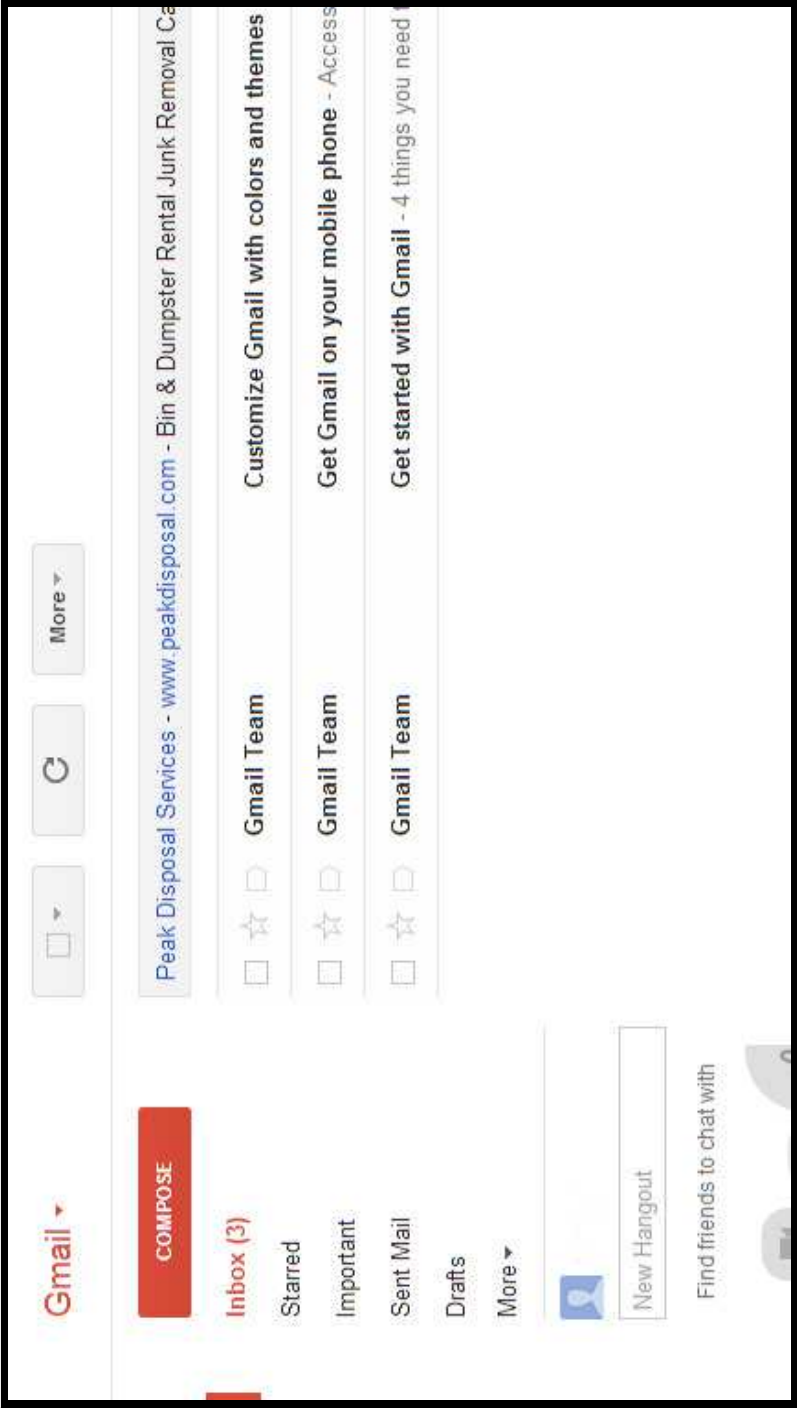


# Google Docs

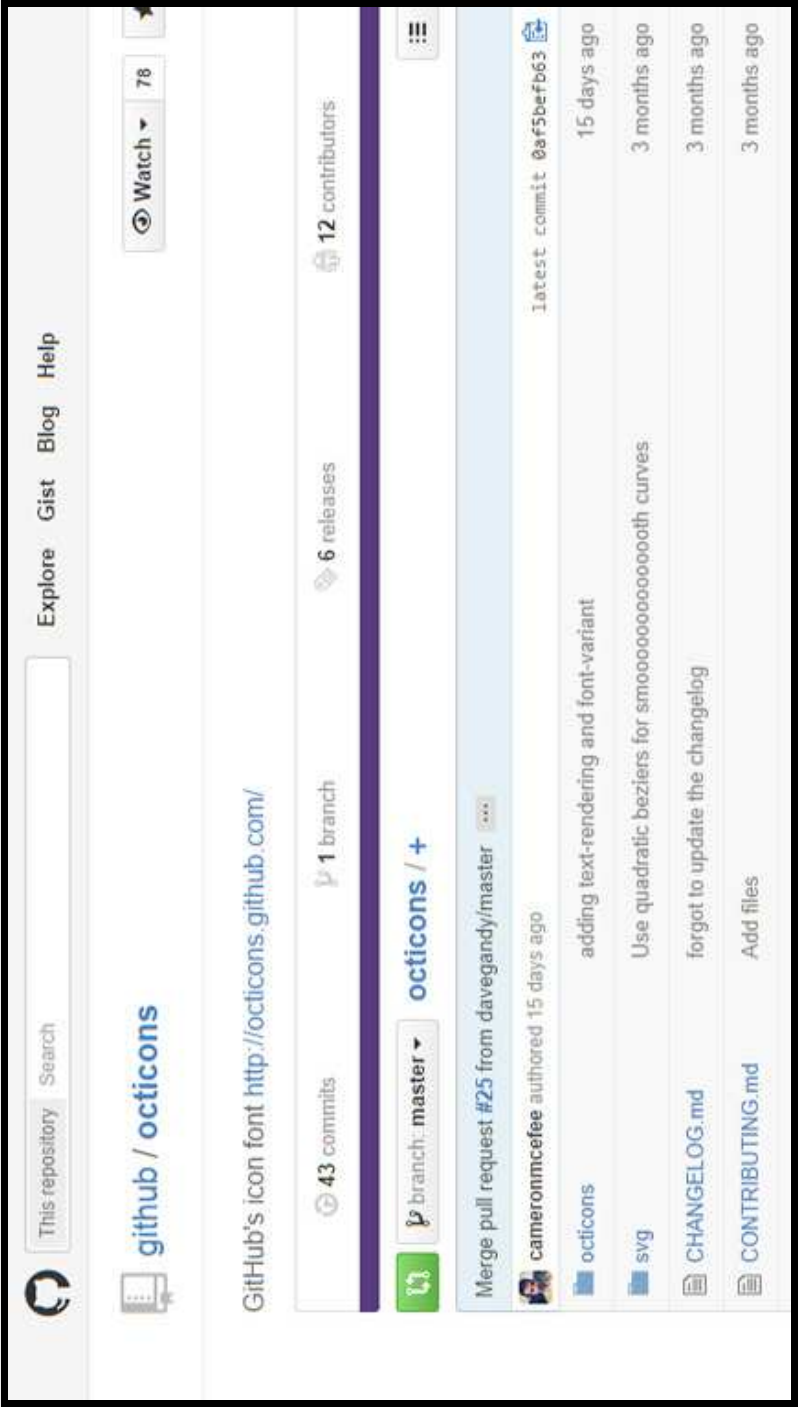




# Webmail (gmail)



# GitHub



# Architectures

- Sites techniquement identiques (client)
- Acquisition / Traitement / Stockage des données différents

# Architecture classique

- Le client fait des requêtes synchrones au serveur
- Le serveur web:
  - récupère les données stockées (ressources, BDD)
  - traite les données
  - répond au client
- Le client affiche les données

# Architecture moderne

- AJAX (Asynchronous JavaScript And XML)
- Le client:
  - fait des requêtes asynchrones au serveur
  - récupère et traite ces informations
  - modifie la page avec le résultat
- → Le socle des applications riches (RIA)

# Application riche

- En plus, le client:
  - traite les données
  - stocke les données
  - affiche / modifie le résultat
  - communique avec d'autres clients
  - accède au système
  - etc
- Le serveur:
  - peut pousser des infos au client (temps-réel)

**Welcome HTML5!**

# HTML5, petite histoire

- 1999: HTML4 par le W3C
- 2000-2007: W3C travaille sur XHTML
  - reformulation de HTML4 en XML
- 2004: Création du WHATWG
  - travaille sur un nouveau standard: HTML5
- 2007: W3C adopte HTML5 sous l'impulsion du WHATWG
- Depuis:
  - W3C abandonne XHTML2
  - HTML5 s'est imposé



# HTML5

- Buzzword!
- HTML5 = HTML + API JavaScript + CSS3
- HTML5: une bonne grosse doc'
- Présentation de HTML5 par W3C
- Permet la création d'application riches!

**HTML5: HTML!**

# HTML, quoi de neuf?

- Simplifie la création de documents web
- Ajout de sémantique
- Meilleurs formulaires
- Améliore l'accessibilité des données

# Structure / règles simplifiées

- XML moins strict
- Moins de balises et d'attributs obligatoires
- Attributs booléens sans valeur, autorisé

```
<input type="checkbox" checked>
```

```
<p contenteditable>Texte éditable</p>
```

- Un élément inline peut contenir des éléments blocs

```
<a href="#">
```

```
<h4>Titre</h4>
```

# Doctype XHTML1



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
```

# Doctype HTML5



```
<!doctype html>
```

- Plus de versions
- Rétro-compatible
- Toujours indispensable

# Structure simplifiée

Document valide HTML5

```
<!doctype html>
```

```
<html lang="fr">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Titre</title>
```

```
</head>
```

```
<body>
```

```
<p>Contenu</p>
```

```
</body>
```

```
</html>
```

# Nouveaux éléments

- Nouvelles balises sémantiques
- Limite les `<div id="">`
  - `section`, `main`, `article`
  - `header`, `footer`
  - `nav`
  - `aside`
  - `figure`, `figcaption`
  - `time`

# Nouveaux éléments

- Nouvelles balises applicatives:
  - `progress`, `meter`
  - `menu`, `command`
  - `keygen`



# Nouveaux formulaires

- Meilleure intégration dans les apps
- Aide l'utilisateur, notamment sur mobile
- Pré-validation client incluse
- Nouveaux `<input type="..." />`
  - `number`
  - `email`, `tel`, `url`
  - `date`, `time`, `datetime-local`
  - `range`
  - `color`
- Nouveaux attributs:
  - `placeholder`, `autofocus`
- Moins besoin de JavaScript
- Native HTML forms

# Attributs personnalisés

- Stockage de données dans le DOM
- `data-attributes`, `data-*`

```
<div data-info="8"></div>
```

# ARIA

- Accessible Rich Internet Applications
- Rendre des composants accessibles
- Ajoute de la sémantique

```
<ol role="tablist">
```

```
<li id="ch1Tab" role="tab">Chapter 1</li>
```

```
<li id="ch2Tab" role="tab">Chapter 2</li>
```

```
</ol>
```

```
<div>
```

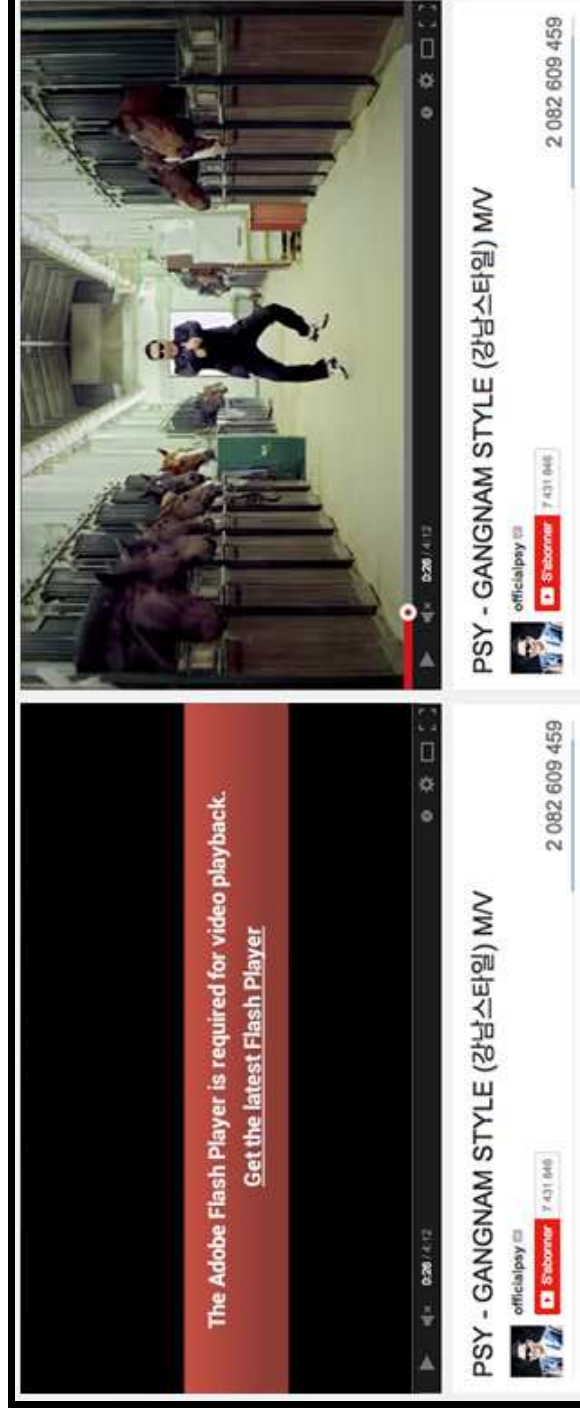
```
<div id="ch1Panel" role="tabpanel" aria-label="Chapter 1">
```

```
<div id="ch2Panel" role="tabpanel" aria-label="Chapter 2">
```

```
</div>
```

# Nouvelles fonctionnalités multimédias

- Sons <audio>
- Vidéos <video>
- Plus de dépendance
- Avant, utilisation de Adobe Flash



**HTML5: CSS!**

# CSS3

- Combo gagnant avec HTML5
- Suite logique du langage CSS

# CSS3, petite histoire

- 1996: CSS1
  - Sélecteurs simples, positionnements basiques, tailles, marges, couleurs, polices, etc.
- 1998: CSS2
  - Abandonnée, car trop complexe
- 2011: CSS2.1
  - Version corrigée de CSS2
- Depuis 1999: CSS3
  - Standard fragmenté en modules
  - Environ 50!
- CSS4 ne verra jamais le jour

# Modules CSS3

- Current work W3C
- Chaque module a sa propre vie
- Chaque module a sa propre version:
  - Sélecteurs Niveau 3, Niveau 4
  - Masques Niveau 1
- Plusieurs modules déjà en recommandation



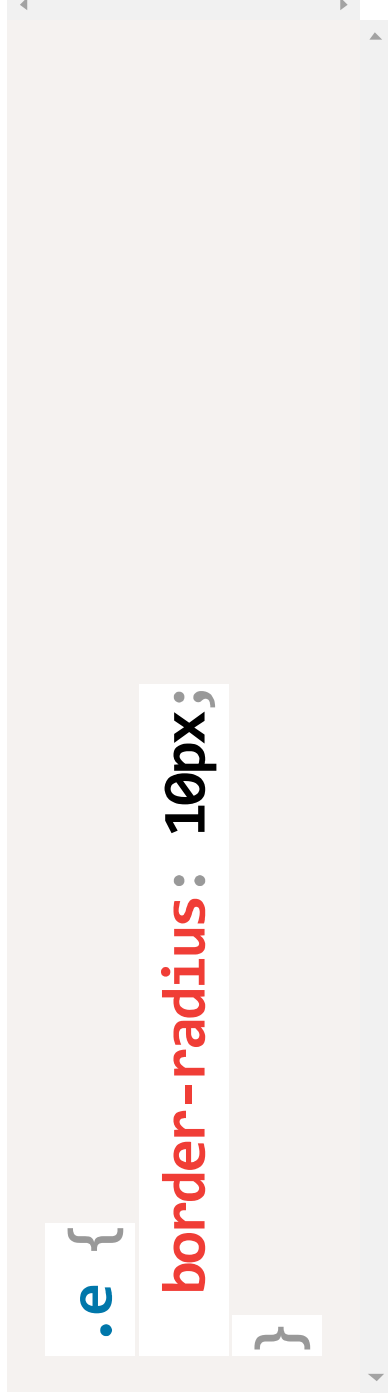
# Sélecteurs CSS3

- Premier module en Recommandation
- Sélecteurs étoffés: 50+ sélecteurs, pseudo-classes, pseudo-éléments
- Ciblage précis des éléments
- Limite les **class** ou **id**
- Sélecteurs d'état de formulaires
- Selectors Test

# Effets graphiques

- Ombres, transparence, couleurs, arrondis, etc.
- Simplification de nombreux effets
- Limiter le nombre d'images
- Maintenance plus facile

# Coins arrondis



- Crée des coins arrondis
- Limite les images
- Simplifie le HTML et le CSS

# Opacité et transparence

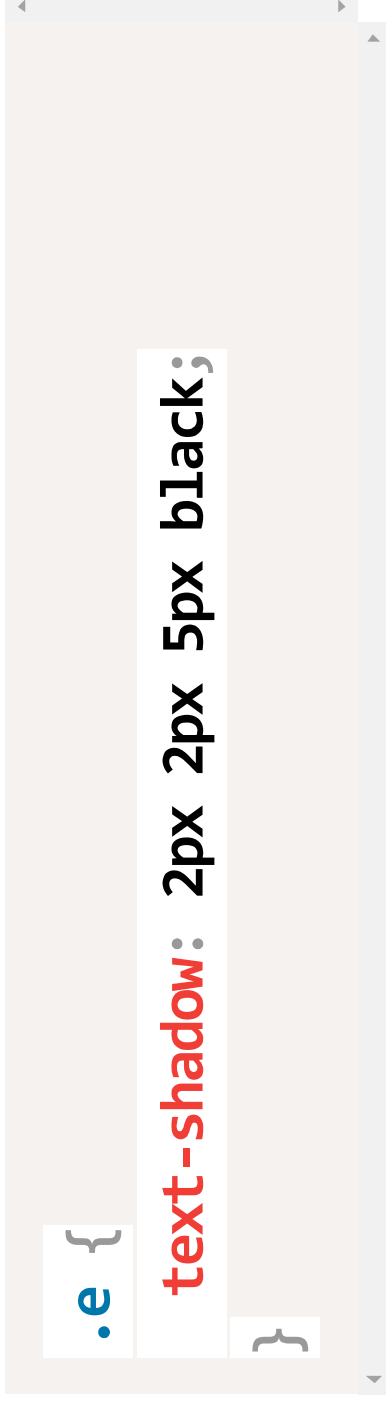
```
.e {  
  opacity: 0.5;  
}
```

- Rends un élément transparent

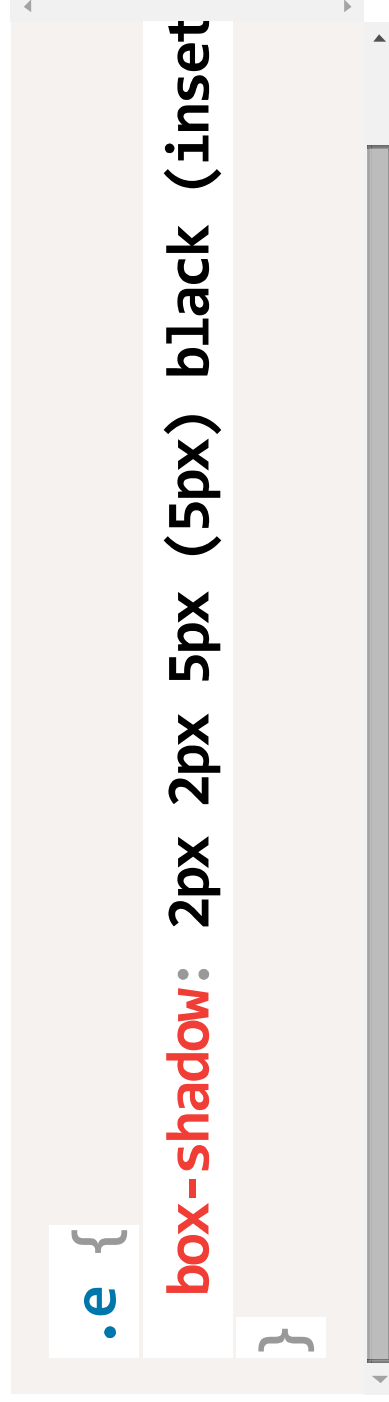
```
.e {  
  color: rgba(235, 162, 53, .5);  
}
```

- Applique une couleur transparente

# Ombres

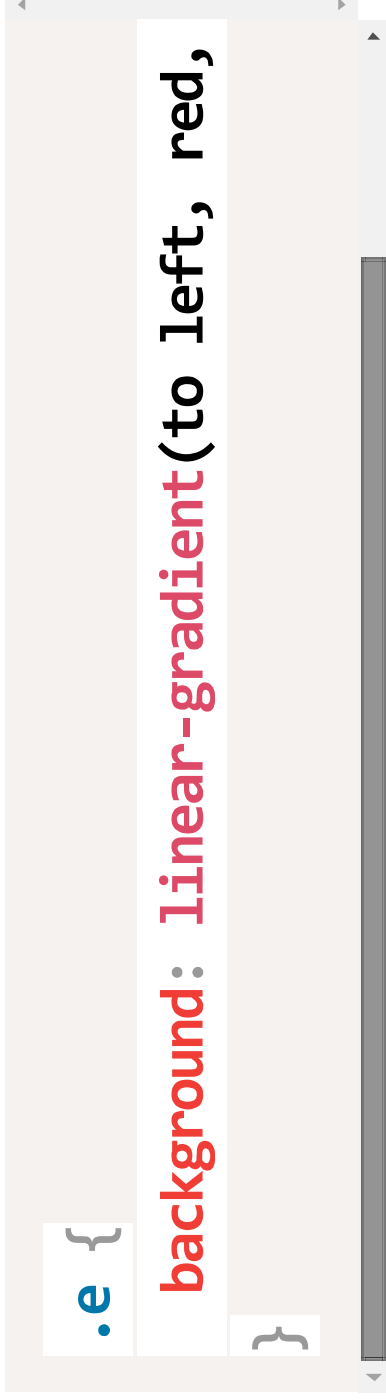


- Ajoute des ombres sur texte



- Ajoute des ombres sur boîtes

# Dégradés



- Dégradés linéaires ou radiaux
- Simplifie grandement l'usage d'images
- Menu sans images
- [Web 2.0 Logo Creatr](#)

# Arrières-plans

- Taille, origin, rognage
- -size, -origin, -clip
- Arrière-plans étirables

```
.e {  
    background-image: url('img.jpg');  
    background-size: cover;  
}
```

# Positionnement

- Nouveaux modes de positionnements
- Flexbox!

```
.e {  
    display: flex;  
    flex-direction: row-reverse;  
    justify-content: space-around;  
}  
.e-item {  
    flex: 1;  
}
```



# Transitions / Animations

- **Transitions:** animer d'un état A à B

```
.e {  
  width: 100px;  
  transition: width 2s;  
}  
.e:hover {  
  width: 200px;  
}
```

- Menu avec transitions

# Transitions / Animations

- **Animations:** animer plusieurs états

```
.e {  
  animation: taille 1s 2 alternate;  
}  
@keyframes taille {  
  from { width: 100px }  
  50% { width: 300px }  
  to   { width: 200px }  
}
```

- Carroussel galerie photo
- Effet de rebond

# Webfonts

- Polices personnalisées

```
@font-face {  
  font-family: 'myFont';  
  src: url(myFont.woff);  
}  
  
h1 {  
  font-family: myFont;  
}
```

- [Google WebFonts](#) et d'autres

# Transformations

```
.e {  
  transform: rotate(45deg) scale(.5);  
}
```

- Transforme un élément dans l'espace
- Animable!

# Transformations 3D

```
.e {  
  transform: rotateY(45deg) rotateX(15deg);  
}
```

- Transforme en 3D
- Animable!
- Flip-card
- Effet overflow 3D
- Rubik's cube

# Filtres

```
.e {  
  filter: blur(2px) grayscale(1);  
}
```

- Processing sur les éléments
- Animable!
- Galerie photo

# Media-queries

- Media depuis CSS2
- CSS3: Requête de média
- Adapter en fonction de critères

```
@media (max-width: 400px) {  
  .e {  
    background: yellow;  
  }  
}
```

# Responsive Web Design!

- Philosophie de développement
- [Showcase mediaqueri.es](http://Showcase.mediaqueri.es)



# HTML5: API JavaScript!

# De nombreuses API...

- Interactions avancées
- Stockage
- Accès système

# Native drag and drop

- Attribut **draggable**
- Nouveaux événements
- Drag&Drop demo

# Géolocalisation

- Coordonnées géographiques de l'utilisateur
  - Basé sur IP
  - Basé sur puce GPS

```
navigator.geolocation.getCurrentPosition(function(position) {  
    console.log(position.coords.latitude, position.coords.longitude);  
});
```

- Géolocalisation démo

# Device Orientation

- Position du device dans l'espace
  - Angles alpha, beta, gamma

```
window.addEventListener('deviceorientation',  
  var a = event.alpha; // autour de Z  
  var b = event.beta;  // autour de X  
  var g = event.gamma; // autour de Y  
, false);
```

- Démo

# Web Storage

- **localStorage**, **sessionStorage**
- Stockage persistent dans le navigateur (seulement pour la session pour **sessionStorage**)
- Clé / valeur simples

```
localStorage.setItem('value', value);  
localStorage.getItem('value');
```

- Démo

# IndexedDB

- Stockage persistant dans le navigateur
- Base de données indexée
- Clé / valeur complexes
- Démo

# Appcache

- Cache Manifest
- Gestion précise du cache

```
<html manifest="cache.appcache">
```

```
CACHE MANIFEST
```

```
# version 1.0.0
```

```
CACHE:
```

```
/html5/src/logic.js
```

```
/html5/src/style.css
```

```
/html5/src/background.png
```



# File API (desktop drag in, drag out)

- Manipulation de fichiers coté client
- Drag & Drop depuis le système
- Démo

# WebWorkers

- Lancer des scripts en parallèle
- Threads
- Performance
- Démo

# Graphiques

- **canvas** pour du dessin matriciel (via JavaScript)
  - Démo simple
  - Video Explosion
  - Pablo The Flamingo
- SVG inline pour du dessin vectoriel
  - Démo simple
  - Coffee Maker
- WebGL (canvas 3D)
  - Randonnée 3D
  - Assassin's Creed Pirates

# WebSockets

- Communication full-duplex, bi-directionnelle
- Le serveur et le client peuvent envoyer des données
- Web temps-réel!
- Pad multiutilisateur
- Flappy Bird temps-réel multijoueur

# Environnement client

- **Graphique**
  - Canvas (WebGL), SVG, CSS, RWD, etc.
- **Contenu**
  - Elements d'UI, Offline, AppCache, History, etc.
- **Données**
  - IndexedDB, WebStorage, WebSockets, WebRTC, etc.
- **Système**
  - Géolocalisation, FileAPI, Battery, SMS, etc.

# Applications

- Création d'application Full HTML5
- Applications mobiles
- Et même, des OS!

# Applications mobiles

## Développement natif

- iOS (Objective C, Swing)
- Android (JAVA)
- Windows Phone (C#)

## Développement hybride

- PhoneGap
- Sencha Touch
- jQuery Mobile

## Développement mobile (HTML5/JS)

- Windows Phone & Windows
- Firefox OS
- Tizen