

# LE WEB

Yann Rotella

Université de Versailles Saint Quentin en Yvelines  
Université Paris-Saclay

22 janvier 2024



HISTORIQUE

LES URLS

HTML

CASCADING STYLE SHEETS

# BREF HISTORIQUE

- ▶ 1969 : ARPAnet, initialement réseau militaire décentralisé qui évolue vers les universités
- ▶ 1972 : Les mails
- ▶ 1991 : apparition du Web

# LE RÉSEAU

- ▶ Décentralisé
- ▶ Communication par paquets
- ▶ Réseaux reliés, réseaux autonomes
- ▶ Ensemble de protocoles pour les transferts de données : mails, WWW, messagerie instantanée, Peer-to-peer, streaming, VisioConf,...

# WORLD WIDE WEB

- ▶ Une application d'internet
- ▶ 1989 par Tim Berners-Lee
- ▶ Système hypertexte (documents avec des liens)
- ▶ Puis images, navigateurs Web, dans les années qui suivent

# DANS LES DÉTAILS

- ▶ hôtes, serveurs, clients
- ▶ Ressources (publiques, privées, locales)
- ▶ TCP/IP, HTTP, TLS
- ▶ URL, HTML
- ▶ Web 2.0 : Modifs, requêtes, temps réel,...

HISTORIQUE

LES URLS

HTML

CASCADING STYLE SHEETS

# LES URLS

- ▶ Uniform Resource Locator. RFC 3986
- ▶ Une chaîne de caractères, Identifiant
- ▶ Hyperliens
- ▶ Affichée dans le navigateur

## FORMAT D'UNE URL

`https :// usr:mdp@ www.ex.fr :N_port /path/... page.html`

# FORMAT D'UNE URL

https :// usr:mdp@ www.ex.fr :N\_port /path/... page.html

Protocole

# FORMAT D'UNE URL

https :// usr:mdp@ www.ex.fr :N\_port /path/... page.html

Protocole

Séparateur

# FORMAT D'UNE URL

https :// usr:mdp@ www.ex.fr :N\_port /path/... page.html

Protocole      Séparateur      Authentication

# FORMAT D'UNE URL

https :// ~~usr:mdp@~~ www.ex.fr :N\_port /path/... page.html

Protocole      Séparateur      Authentication

The diagram shows a URL 'https://usr:mdp@www.ex.fr:N\_port/path/...page.html'. The 'https' part is underlined in blue, with a blue line pointing to the label 'Protocole'. The '://' part is underlined in blue, with a blue line pointing to the label 'Séparateur'. The 'usr:mdp@' part is crossed out with a red 'X', and a red line points from this area to the label 'Authentication'.

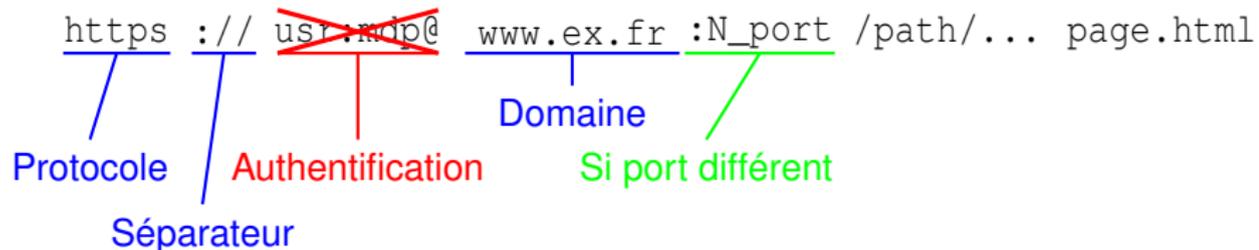
# FORMAT D'UNE URL

https :// ~~usr:mdp@~~ www.ex.fr :N\_port /path/... page.html

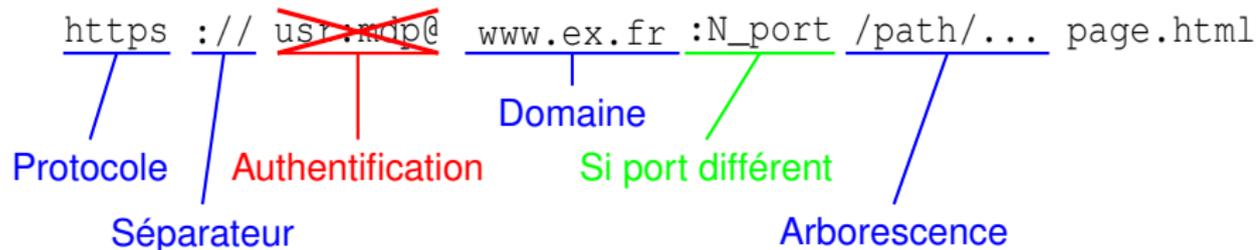
Protocole      Séparateur      Authentication      Domaine

The diagram illustrates the components of a URL. The URL is shown as 'https://usr:mdp@www.ex.fr:N\_port/path/...page.html'. The 'https' part is underlined in blue and labeled 'Protocole'. The '://' part is underlined in blue and labeled 'Séparateur'. The 'usr:mdp@' part is crossed out with a red 'X' and labeled 'Authentication'. The 'www.ex.fr' part is underlined in blue and labeled 'Domaine'. The ':N\_port' part is not underlined. The '/path/...page.html' part is not underlined.

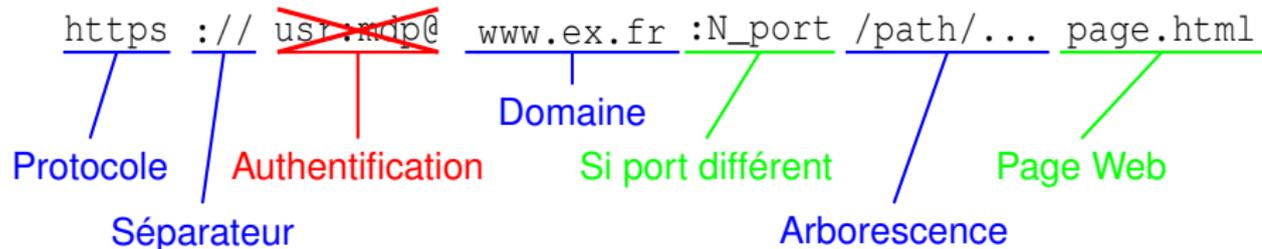
# FORMAT D'UNE URL



# FORMAT D'UNE URL

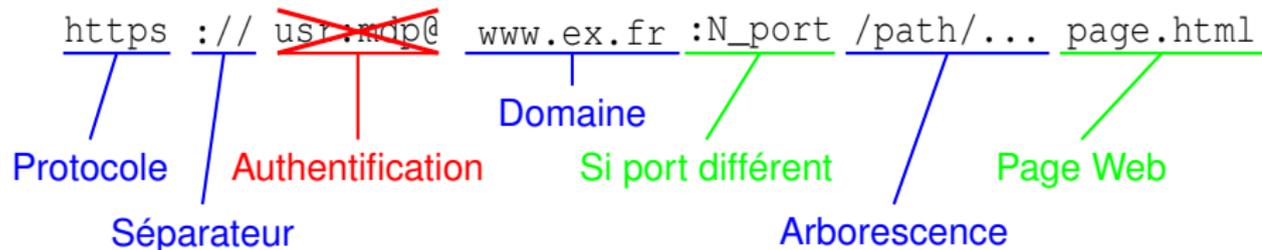


# FORMAT D'UNE URL



`?q=req1&q2=req2 #id`

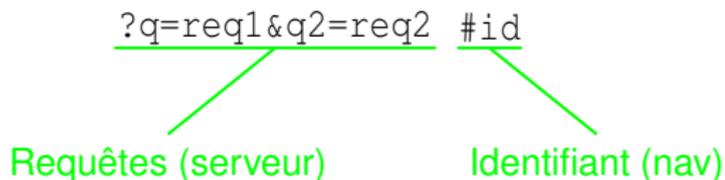
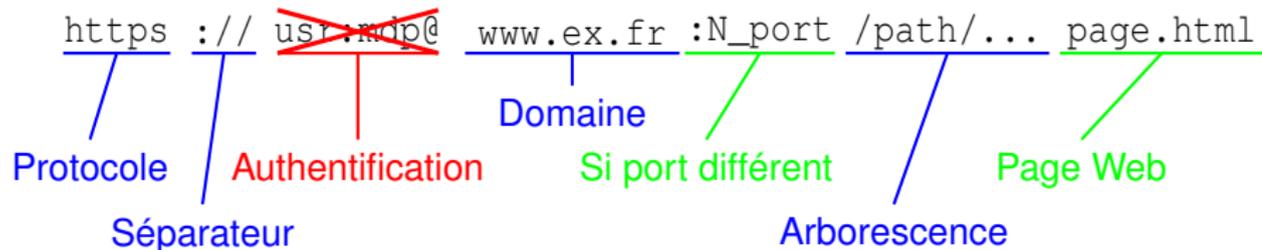
# FORMAT D'UNE URL



?q=req1&q2=req2 #id

Requêtes (serveur)

# FORMAT D'UNE URL



# URL RELATIVES

Naviguer dans le document :

- ▶ ./ dossier actuel
- ▶ ../ dossier parent
- ▶ / dossier racine

HISTORIQUE

LES URLS

**HTML**

CASCADING STYLE SHEETS

# LES BALISES

HTML : Hypertext Markup Language

# LES BALISES

HTML : Hypertext Markup Language

```
<tag> Bienvenue sur mon site </tag>
```

# LES BALISES

## HTML : Hypertext Markup Language

```
<tag> Bienvenue sur mon site </tag>
```

- ▶ Les balises (<tag> et </tag>) délimitent le **contenu**
- ▶ On a toujours une balise ouvrante qui correspond à une balise fermante
- ▶ Attention à l'imbrication ! Et à la bonne tabulation, naturellement...
- ▶ Raccourci d'une balise sans contenu : 

```
<tag/>
```

# LES ATTRIBUTS

Les balises peuvent ou doivent avoir des attributs.

```
<tag att1="une valeur" att2='une autre' att3=etuneautre>  
    contenu  
</tag>
```

# PRODUIRE UN DOCUMENT LISIBLE

```
<html>
  <head>
    <meta charset="utf-8">
    <title> My Title </title>
    <!-- And more (style, script, ...) -->
  </head>
  <body>
    <!-- Le contenu du doc html -->
  </body>
</html>
```

# HEAD

- ▶ `<title>` : titre du document
- ▶ `<meta>` : language, encodage,...
- ▶ `<link>` : quelques liens
- ▶ `<script>` : Script (côté client)
- ▶ `<base>` : pour les URL relatives
- ▶ `<style>` : CSS

# BODY

- ▶ `<section>`, `<nav>`, `<header>`, `<footer>` : **structure**
- ▶ `<p>`, `<h1-6>` : **Niveaux d'importance**
- ▶ `<a>` : **lien**
- ▶ `<img>`, `<audio>`, `<video>`, `<object>` : **médias**
- ▶ `<form>`, `<input>` : **formulaire**
- ▶ `<table>`, `<tr>`, `<td>` : **tableaux**
- ▶ `<strong>`, `<em>` : **gras, italique**
- ▶ `<div>`, `<span>` : **utilisation du CSS**

# LES ATTRIBUTS

- ▶ `id` : identifiant, **Unique**
- ▶ `class` : classe
- ▶ `src` : liens, incorporation de la ressource
- ▶ `href` : liens, relation avec la ressource externe
- ▶ `title` : rajout d'informations
- ▶ `style` : style
- ▶ `onclick`, `onload`, `onmouseover` : événements
- ▶ `data-*`

# LES BONNES PRATIQUES

- ▶ Structure et logique
- ▶ Imbrication et tabulation
- ▶ Déclarer le langage et l'encodage
- ▶ Utiliser les standards et les validateurs
- ▶ Séparation style et contenu (CSS)

# LES FORMULAIRES

`<form>` **et** `<input>` :

`text`, `radio`, `checkbox`, `password`, `file`, `email`, `submit`,  
`button`, `image`, `date`, `time`, `number`, `range`, `color`, `tel`,  
`url`

`<textarea>`, `<select>`, `<datalist>`, `<keygen>`, `<output>` :

# CONTROLLER LES FORMULAIRES

- ▶ `required` : remplissage
- ▶ `pattern` : `RegExp` !
- ▶ `placeholder` : suggestion
- ▶ et d'autres !

# QUELQUES RÉFÉRENCES

- ▶ RFC URL
- ▶ Pour s'amuser
- ▶ The World Wide Web Consortium
- ▶ w3 validator, CSS validator, checklink
- ▶ HTML-Mozilla dev, HTML w3
- ▶ HTML forms Mozilla, HTML forms w3

BUT THIS IS THE VERY BEGINING...

Road Map

HISTORIQUE

LES URLS

HTML

CASCADING STYLE SHEETS

# LA RÈGLE DE BASE

On sépare la **sémantique** de la **présentation**.

# SYNTAXE CSS

- ▶ `selecteur {propriete1: valeur1; propriete2: valeur2; }`
- ▶ `/* un commentaire*/`
- ▶ `@import "style2.css"`
- ▶ `@media!`

# SÉLECTEURS CSS

- ▶ `tag {...}`
- ▶ `tag.class {...}`
- ▶ `# id {...}`
- ▶ `tag:pseudoclass {...}` (`visited`, `hover`, ...)
- ▶ `tag[att=val] {...}`

# COMBINER LES SÉLECTEURS

- ▶ `selecteur1, selecteur2 {...}`
- ▶ `parent child {...}`
- ▶ `parent > child {...}`
- ▶ `sister brother {...}`
- ▶ `sister + brother {...}`

# RÈGLES EN CASCADE

Du moins prioritaire au plus prioritaire :

- ▶ Style par défaut du navigateur
- ▶ Feuilles de style intégrées dans le `<head>`
- ▶ Style en ligne (On évite !)
- ▶ Styles définis par l'utilisateur

# SPÉCIFICITÉ

La règle en cascade est appliquée à spécificité égale :

- ▶ nombre d'ID
- ▶ nombre de classes et pseudo-classes
- ▶ nombre d'éléments

# UTILISATION

- ▶ Animations, transitions
- ▶ Responsive design (media queries)
- ▶ Layouts
- ▶ Framework Frontend ([Bootstrap](#), [PureCSS](#),...)

# LE BOX MODEL

## BLOCK

Prend toute la largeur de son conteneur, `width` et `height` toujours respectés. `padding`, `margin` et `border` repoussent les autres. (`<h1>`, `<h2>`, `<p>` par défaut.

# LE BOX MODEL

## BLOCK

Prend toute la largeur de son conteneur, `width` et `height` toujours respectés. `padding`, `margin` et `border` repoussent les autres. (`<h1>`, `<h2>`, `<p>` par défaut.

## INLINE

Un `inline` revient à la ligne et ne prend que la place qu'elle occupe. `width` et `height` ne s'appliquent pas. Verticalement, ne repousse pas, horizontalement, repousse. `<a>`, `<span>` par défaut.

# LE BOX MODEL

## BLOCK

Prend toute la largeur de son conteneur, `width` et `height` toujours respectés. `padding`, `margin` et `border` repoussent les autres. (`<h1>`, `<h2>`, `<p>` par défaut.

## INLINE

Un `inline` revient à la ligne et ne prend que la place qu'elle occupe. `width` et `height` ne s'appliquent pas. Verticalement, ne repousse pas, horizontalement, repousse. `<a>`, `<span>` par défaut.

## INLINE-BLOCK

Ne reviennent pas à la ligne, mais il peut y avoir plusieurs sur une même ligne.

# UTILISATION

On peut utiliser la propriété `display`.

# UTILISATION

On peut utiliser la propriété `display`.

- ▶ Un bloc ne va pas dans un inline
- ▶ On s'aide du DevTools d'un navigateur par exemple celui de [Firefox](#)
- ▶ [Box Model](#)

# UTILISATION

On peut utiliser la propriété `display`.

- ▶ Un bloc ne va pas dans un inline
- ▶ On s'aide du DevTools d'un navigateur par exemple celui de [Firefox](#)
- ▶ [Box Model](#)

## POSITIONNEMENT

de haut en bas, de gauche à droite, et `float`, mais aussi `static`, `relative`, ... [Floats](#) et [Position](#).

# FLEXBOX

- ▶ Centrer
- ▶ Optimiser l'espace

# FLEXBOX

- ▶ Centrer
- ▶ Optimiser l'espace

`flex-direction`, `flex-wrap` et leur abrégé `flex-flow`

# FLEXBOX

- ▶ Centrer
- ▶ Optimiser l'espace

`flex-direction`, `flex-wrap` et leur abrégé `flex-flow`

FlexBox

# CSS GRID LAYOUT

- ▶ Le modèle le plus récent
- ▶ Responsive
- ▶ Meilleur code HTML
- ▶ `display: grid`

# QUELQUES PROPRIÉTÉS CSS

- ▶ `grid, inline-grid`
- ▶ `grid-template-rows, grid-template-columns`
- ▶ `grid-row, grid-column`
- ▶ `grid-gap's`
- ▶ `align-item` **et** `justify-item`

# QUELQUES PROPRIÉTÉS CSS

- ▶ `grid, inline-grid`
- ▶ `grid-template-rows, grid-template-columns`
- ▶ `grid-row, grid-column`
- ▶ `grid-gap's`
- ▶ `align-item` **et** `justify-item`

Attention à la compatibilité

# UTILISATION

- ▶ Nommer les emplacements de la grille dans le CSS
- ▶ classiques px, %, et fr
- ▶ min-content, max-content, auto
- ▶ rowspan et colspan en HTML deviennent / span \*
- ▶ repeat()
- ▶ grid-auto-flow

# UTILISATION

- ▶ Nommer les emplacements de la grille dans le CSS
- ▶ classiques px, %, et fr
- ▶ min-content, max-content, auto
- ▶ rowspan et colspan en HTML deviennent / span \*
- ▶ repeat()
- ▶ grid-auto-flow

Et surtout : @media + CSS Grid!