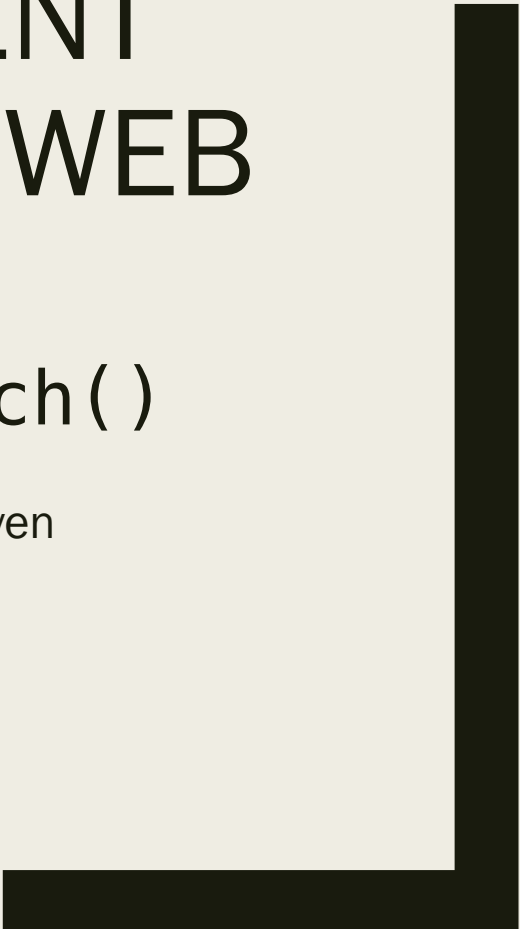




DÉVELOPPEMENT D'APPLICATIONS WEB

COURS 3: CSS

Enseignante: NGUYEN Thi Minh Tuyen



Plan du cours

1. Les avantages de CSS
2. Comprendre la structure du document
3. Règles de style d'écriture
4. Attacher des styles au document HTML
5. Héritage
6. La cascade
7. Le modèle de boîte
8. Unités de mesure CSS

CSS

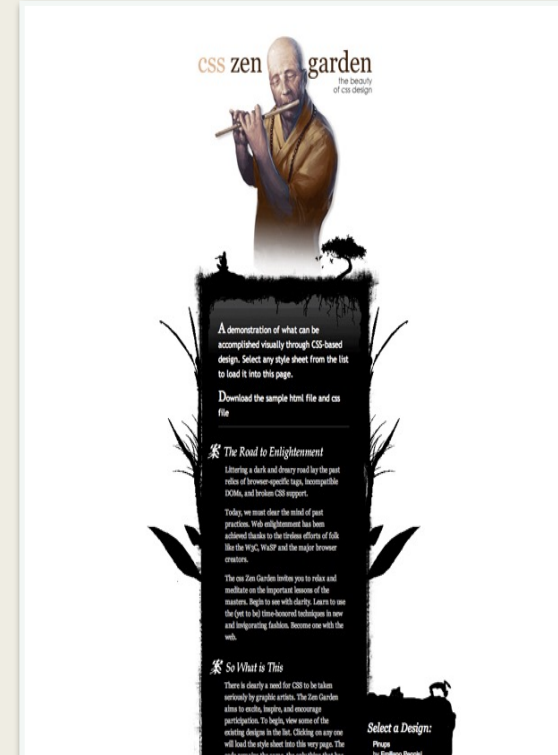
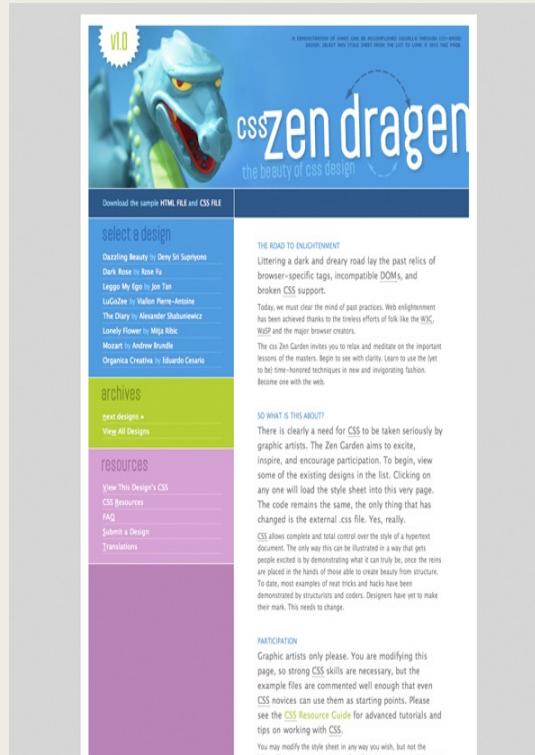
- Cascading Style Sheets
- Norme W3C
- Définit la présentation des documents écrits en HTML et d'autres langages XML.

Les avantages de CSS

- Contrôle de type et de mise en page précis
- Moins de travail:
 - *Modifier l'apparence de site entier en une seule modification.*
- Accessibilité:
 - *Lorsque la présentation est géré par CSS, vous pouvez baliser votre contenu de manière significative, le rendant plus accessible pour les appareils non visuels ou mobiles.*
- Flexibilité:
 - *le même balisage HTML peut apparaître de manière très différente*

Style séparé de la structure

- Ces pages ont exactement la même source HTML mais des feuilles de style différentes:



Source: (csszengarden.com)

Comment fonctionnent les feuilles de style ?

1. Commence avec un document marqué (comme HTML, mais pourrait être un autre langage de balisage XML).
2. Rédige des styles pour définir l'apparence des éléments à l'aide de la syntaxe CSS.
3. Attache les styles au document (il y a plusieurs façons).
4. Le navigateur affiche le document en suivant vos instructions lors.

Règles de style

- Chaque règle sélectionne un élément et déclare son affichage.

```
h1 { color: green; }
```

- Cette règle sélectionne tous les éléments h1 et déclare qu'ils doivent être verts.

```
strong { color: red; font-style: italic; }
```

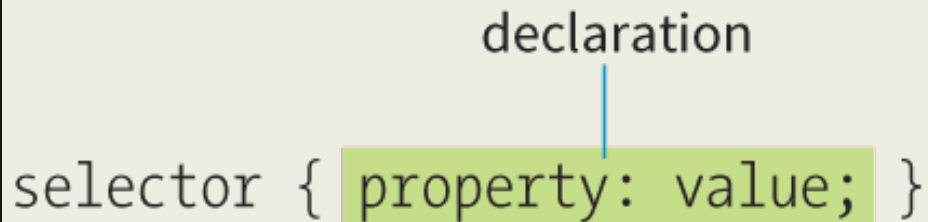
- Cette règle sélectionne tous les éléments en ligne forts et déclare qu'ils doivent être en rouge et en italique.

Structure de règle de style

- Une règle de style est composée d'un sélecteur d'une déclaration.
- La déclaration est une ou plusieurs paires propriété / valeur.

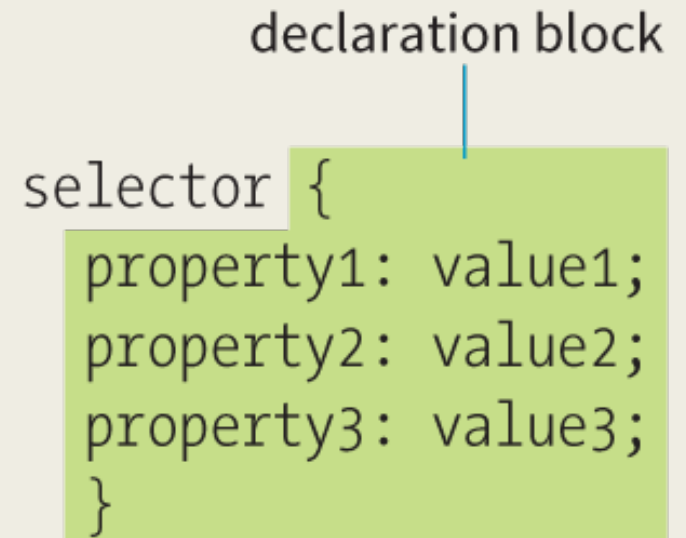
declaration

selector { property: value; }

A diagram illustrating a single declaration within a selector block. The text "selector { property: value; }" is shown. A light green rectangular box highlights the text "property: value;". A blue vertical line points from the word "declaration" above to the highlighted box.

declaration block

selector {
property1: value1;
property2: value2;
property3: value3;
}

A diagram illustrating a declaration block within a selector. The text "selector {
property1: value1;
property2: value2;
property3: value3;
}" is shown. A light green rectangular box highlights the entire block of declarations. A blue vertical line points from the words "declaration block" above to the highlighted box.

Sélecteurs

Il existe de nombreux types de sélecteurs. Voici juste deux exemples:

```
p {property: value;}
```

Sélecteur de type d'élément: sélectionne tous les éléments de ce type (**p**) dans le document.

```
#intro {property: value}
```

Le sélecteur ID (indiqué par le symbole **#**) sélectionne par valeur ID. Dans l'exemple, un élément avec un id de "intro" serait sélectionné.

Déclarations [1]

- La déclaration est composée d'une paire propriété/valeur mise entre accolades {}:

```
selector { property: value; }
```

- Exemple:

```
h2 { color: red;  
    font-size: 2em;  
    margin-left: 30px;  
    opacity: .5;  
}
```

Déclarations [2]

- Termine chaque déclaration par un point-virgule ; pour la conserver séparément de la déclaration suivante.
- Les espaces sont ignorés. Vous pouvez donc empiler des déclarations pour les rendre plus lisibles.
- Les propriétés sont définies dans les spécifications CSS.
- Les valeurs dépendent du type de propriété:
 - *Des mesures*
 - *Mots clés*
 - *Valeurs de couleur*
 - ...

Commentaires CSS

```
/* comment goes here */
```

- Le contenu entre / * et * / sera ignoré par le navigateur.
- Utile pour laisser des notes ou des étiquettes de section dans la feuille de style.
- Peut être utilisé dans les règles pour cacher temporairement les déclarations de style dans le processus de conception.

Ajout de styles au document

Il existe trois façons d'attacher une feuille de style à un document:

- **Feuilles de style externes**

- *Un fichier **.css** séparé, texte uniquement, associé au document avec l'élément **link** ou la règle **@import***

- **Feuilles de style intégré**

- *Les styles sont listées dans l'en-tête du document HTML dans l'élément **style**.*

- **« Inline styles »**

- *Les propriétés et les valeurs sont ajoutées à un élément individuel avec l'attribut **style**.*

Feuilles de style externes

- Les règles de style sont enregistrées dans un fichier .css séparé au format texte et sont jointes via **link** ou **@import**.

- Via élément de link :

```
<head>
  <title>Titles are require</title>
  <link rel="stylesheet" href="/path/example.css">
</head>
```

- Via @import:

```
<head>
  <title>Titles are required</title>
  <style>
    @import url("/path/example.css");
    p {font-face: Verdana;}
  </style>
</head>
```

Feuilles de style intégrées

- Les feuilles de style intégrées sont placées dans l'en-tête du document via l'élément de **style**:

```
<head>  
  <title>Titles are required</title>  
  <style>  
    /* style rules go here */  
  </style>  
</head>
```

« Inline Styles »

- Applique une déclaration de style à un seul élément avec l'attribut **style**:

```
<p style="font-size: large;">Paragraph text...</p>
```

- Pour ajouter plusieurs propriétés, sépare-les par des points-virgules ;

```
<h3 style="color: red; margin-top: 30px;">Intro</h3>
```


Structure du document

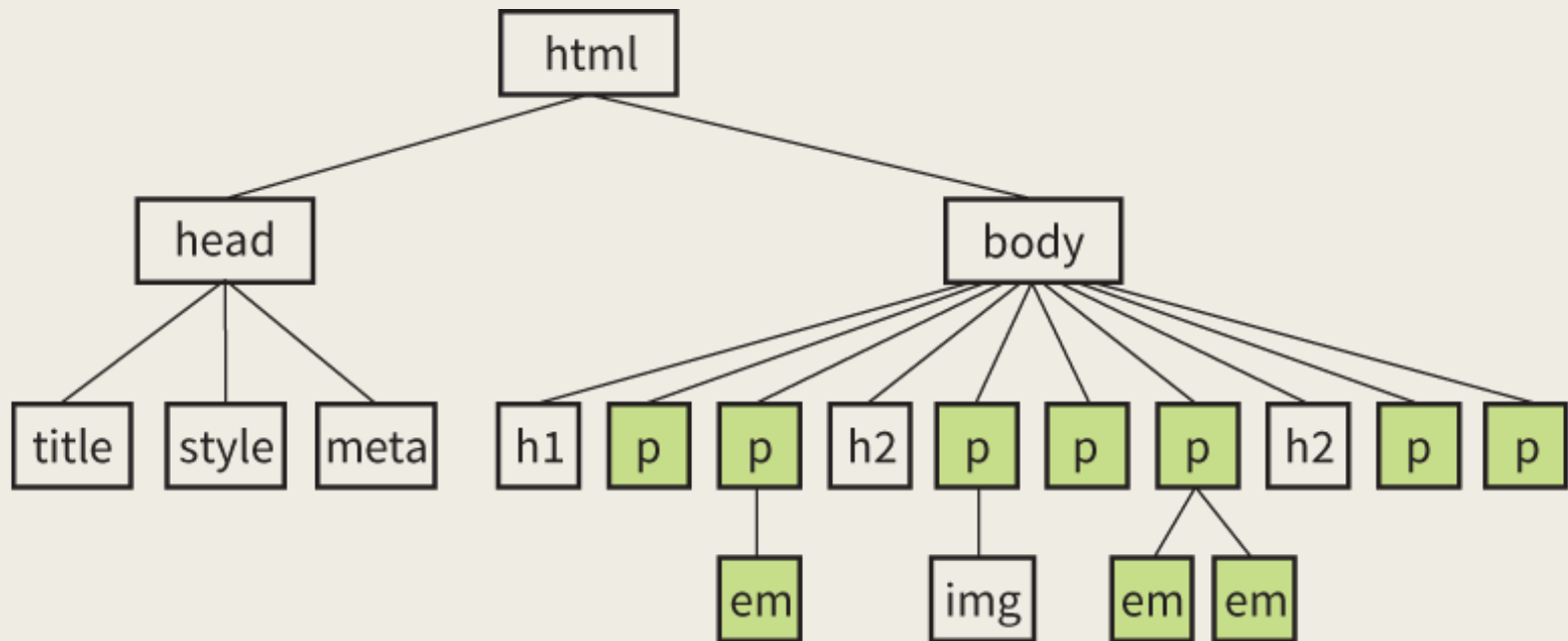
Les documents ont une structure implicite.

Nous donnons certains noms de relations, comme s'il s'agissait d'une famille:

- Tous les éléments contenus dans un élément donné sont ses descendants.
- Un élément directement contenu dans un autre élément est l'enfant de cet élément.
- L'élément conteneur est le parent de l'élément contenu.
- Deux éléments avec le même parent sont frères (siblings).

Héritage [1]

- De nombreuses propriétés appliquées aux éléments sont transmises aux éléments qu'ils contiennent. Ceci s'appelle l'héritage.
- Par exemple, si vous appliquez une police **sans-serif** à un élément **p**, l'élément **em** qu'il contient sera également **sans-serif**:



`p {font-size: large; font-family: sans-serif;}`

Héritage [2]

- Certaines propriétés héritent; D'autres ne le font pas.
- En générale: les propriétés liées au texte héritent; les propriétés liées à la mise en page ne le sont pas.
- Les styles explicitement appliqués à des éléments spécifiques remplacent les styles hérités.
- Vous apprendrez à utiliser l'héritage de manière stratégique pour que vos règles de style restent simples.

Cascade

- La cascade fait référence au système pour résoudre des conflits lorsque plusieurs styles s'appliquent au même élément.
- Les informations de style sont transmises (cascade) jusqu'à ce qu'elles soient remplacées par une règle de style ayant le plus de poids.
- Le poids est considéré en fonction de:
 - *Priorité de la source de la règle de style*
 - *Spécificité du sélecteur*
 - *Ordre de règle*

Cascade: priorité

Les règles de style des sources situées plus haut dans cette liste remplacent les règles des sources énumérées ci-dessous.

1. Tout style marqué comme **!important** par l'utilisateur (pour s'adapter aux paramètres d'accessibilité potentiels)
2. Tout style marqué **!important** par l'auteur (de la page Web)
3. Styles d'auteur (feuilles de style créées lors de la production de sites Web)
4. Styles d'utilisateur (ajoutés par le lecteur)
5. Styles d'agent utilisateur (paramètres par défaut du navigateur)

Cascade: spécificité

- Lorsque deux règles d'une même feuille de style sont en conflit, le type de sélecteur est utilisé pour déterminer quelle règle a le plus de poids.
- Par exemple: les sélecteurs d'ID sont plus spécifiques que les sélecteurs d'éléments généraux.

Cascade: ordre de règle

- Lorsque deux règles ont le même poids, l'ordre des règles est utilisé. La dernière règle apparue va gagner.

```
<style>
```

```
p {color: red;}
```

```
p {color: blue;}
```

```
p {color: green;}
```

```
</style>
```

- Dans cet exemple, les paragraphes seraient verts.
- Les styles peuvent être entrés de feuilles de style externes, de règles de style intégrées et de styles au niveau texte. La règle de style qui est analysée en dernier (celle la plus proche du contenu) s'appliquera.

Modèle de boîte [1]

- Les navigateurs voient chaque élément de la page comme contenu dans une petite boîte rectangulaire. Les éléments de bloc et les éléments au niveau texte participent au modèle de boîte.
- Dans cet exemple, une bordure bleue est ajoutée à tous les éléments.

Cooking with Daniel from Nada Surf

I had the pleasure of spending a crisp, Spring day in Portsmouth, NH cooking and chatting with Daniel Lorca of the band Nada Surf as he prepared a gourmet, sit-down dinner for 28 pals.

When I first invited Nada Surf to be on the show, I was told that Daniel Lorca was the guy I wanted to talk to. The response: "I'm way into it, but I don't want to talk about it, I wanna do it." After years of only having access to their sound check and set, I've been doing a lot of *talking* about cooking with rockstars. To actually cook is true.

Six-hour Salad



Daniel prepared a salad of arugula, smoked tomatoes, tomato jam, and grilled avocado (it's as good as it sounds!). I jokingly called it "6-hour Salad" because that's how long he worked on it. The fresh tomatoes were slowly smoked over woodchips in the grill, and when

was to
k about it, i
of *talking* at

Modèle de boîte [2]

- Le modèle de boîte est la base de la mise en page CSS.
- Appliquez des propriétés telles que **borders**, **margins**, **padding**, et **backgrounds** d'éléments.
- Positionner, déplacer, étendre et réduire les boîtes pour créer des mises en page fixes ou flexibles.

Unités de mesure CSS

CSS fournit diverses façons de spécifier des mesures:

- **Unités absolues**

- *Ont des significations prédéfinies ou des équivalents du monde réel*

- **Unités relatives**

- *Basé sur la taille de quelque chose d'autre, comme la taille de texte par défaut ou la taille de l'élément parent*

- **Les pourcentages**

- *Calculé par rapport à une autre valeur, telle que la taille de l'élément parent*

Unités absolues

À l'exception des pixels, les unités absolues ne sont pas appropriées pour la conception Web:

- **px** pixel
- **in** inches
- **mm** millimeters
- **cm** centimeters
- **q** 1/4 millimeter
- **pt** points (1/72 inch)
- **pc** pica (1 pica = 12 points = 1/6 inch)

Unités relatives

Les unités relatives sont basées sur la taille de quelque chose d'autre:

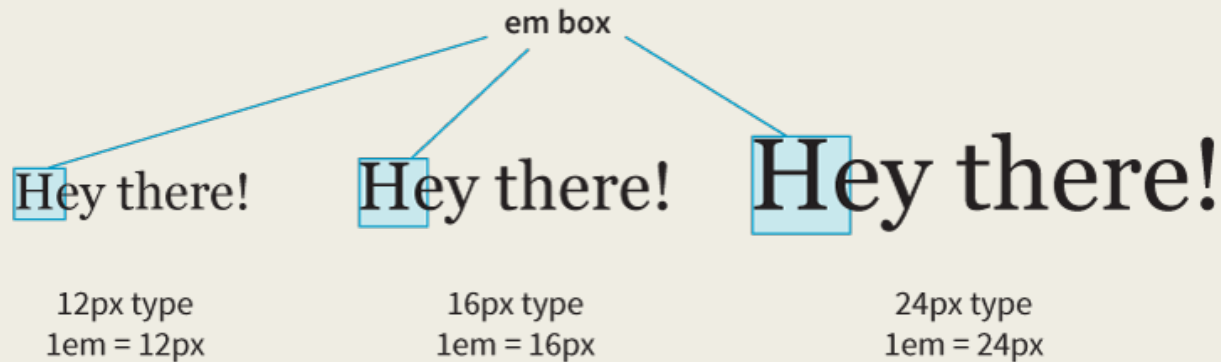
- **em** une unité égale à la taille de la police de l'élément actuel
- **ex** x-height, égale à la hauteur d'un x minuscule
- **rem** root em, égal à la taille de la police de l'élément html
- **ch** zéro largeur, égale à la largeur d'un zéro (0)
- **vw, vh** représentent respectivement un centième de la largeur de la vue (viewport) et un centième de sa hauteur
- **vmin** Unité minimale de la vue (valeur de **vh** ou **vw**, selon la valeur la plus petite)
- **vmax** Unité maximale de la vue (valeur de **vh** ou **vw**, selon la valeur la plus grande)

Unité rem

- L'unité **rem** (**root em**) est basée sur la taille de la police de l'élément html, quel que soit le résultat.
- Par défaut dans les navigateurs modernes: la taille de la police de base est de 16 pixels, donc une unité rem = une unité de 16 pixels.
- Si la taille de la police racine du document change, il en va de même pour la taille d'un rem (ce qui permet de garder les éléments proportionnels).

Unité em

- L'unité **em** est traditionnellement basée sur la largeur d'une lettre majuscule **M** dans la police.
- Lorsque la taille de la police est de 16 pixels, 1em = 16 pixels, 2em = 32 pixels, etc.



Longueurs en pourcentage de la vue (vw / vh)

- Les unités largeur (vw) et hauteur de la fenêtre (vh) sont relatives à la taille de la fenêtre (fenêtre du navigateur):
- $vh = 1/100\text{ème}$ de largeur de la fenêtre
- $vh = 1/100\text{ème}$ hauteur de la fenêtre
- Ils sont utiles pour qu'un élément remplisse la fenêtre ou un pourcentage spécifié de celle-ci. Cette image correspondra à 50% de la largeur et de la hauteur de la fenêtre:

```
img { width: 50vw; height: 50vh; }
```

Outils de développement du navigateur [1]

Les principaux navigateurs ont des outils intégrés qui aident au développement:

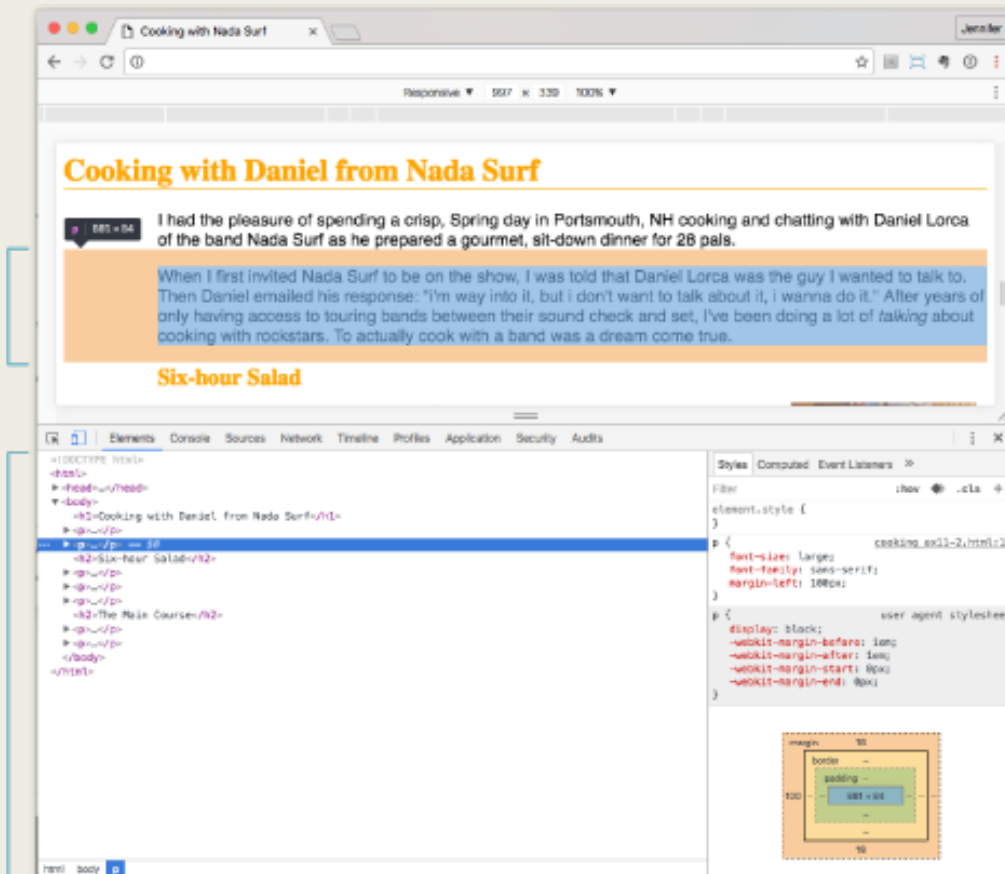
- Inspecteurs HTML, CSS et JavaScript
- Rapports de vitesse du réseau
- Outils d'animation
- Autres fonctionnalités utiles

Outils de développement du navigateur [2]

- Chrome DevTools (View > Developer > Developer Tools)

Elements selected in code are highlighted in the browser view.

HTML source for the page.



All styles that are applied to the selected element.

Margins, borders, and paddings applied to the element.

Question ?