

Landing Page de A à Z

Cours 4

Interactions utilisateurs

Sur une page web, l'utilisateur est amené à interagir avec des éléments comme :

- Remplir un formulaire de contact
- Cliquer sur un lien / bouton

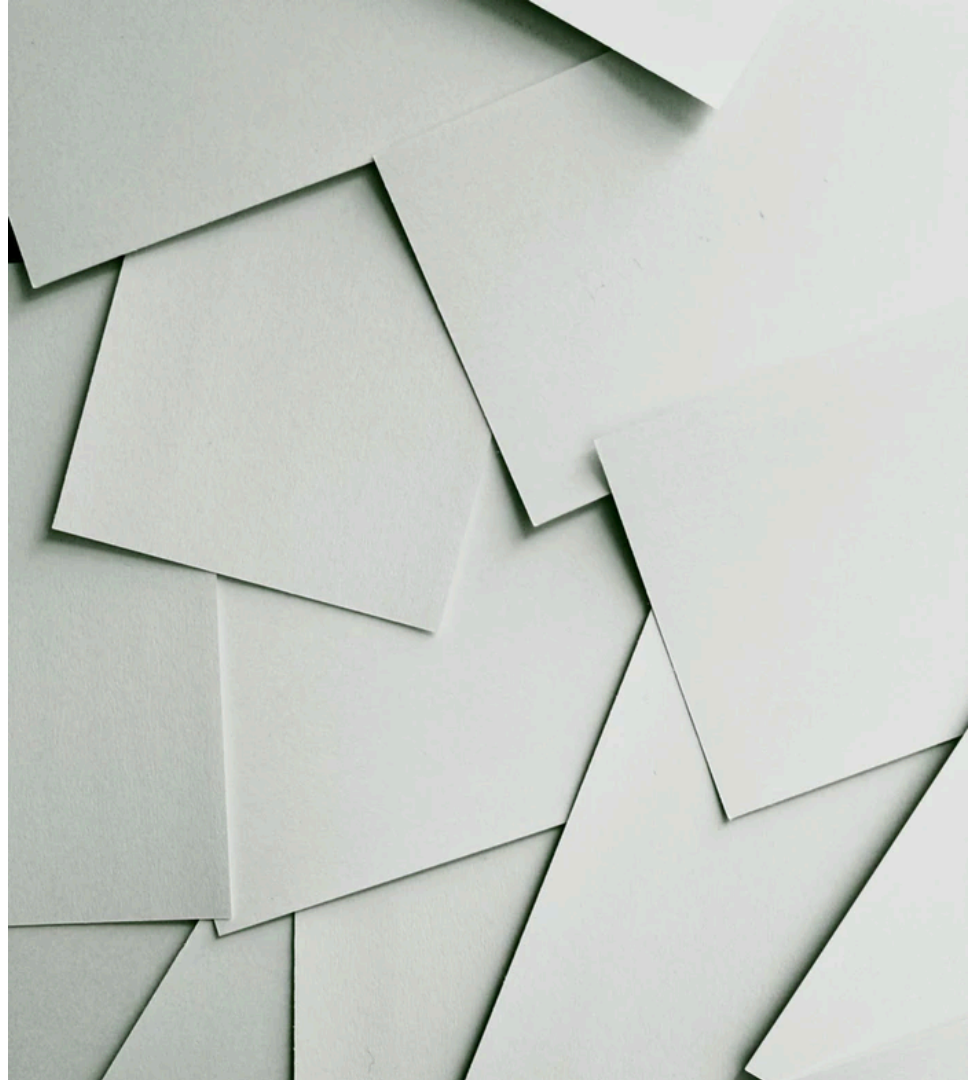
Il est important de savoir comment cela fonctionne et comment gérer ce genre d'interactions en CSS.

Deux Parties

1. Créer un formulaire en HTML/CSS

2. Animer des interactions en CSS

Les formulaires



Pourquoi les formulaires ?

- Nécessaires pour récupérer
 - des **informations de l'utilisateur** : nom, mot de passe, email etc.
 - des **commandes** : clic sur un bouton pour activer ou non une fonctionnalité sur un site etc.
- Incontournables pour n'importe quel site : formulaire de connexion, de contact, d'achat..
- Le formulaire **s'affiche côté client en HTML/CSS**, et **traite les données côté serveur avec PHP** (ou autre langage serveur).

Au commencement, la balise <form>

-

```
<form>
```

```
<!-- my form code here -->
```

```
</form>
```

Exemple d'un formulaire simple

```
<form>
```

```
<input type='text' name='prenom' value='ton prenom'>
```

```
<input type='submit' value='Valider'>
```

```
</form>
```

Que se passe-t-il si on fait 'submit' ?

- Le formulaire envoie les données à une page PHP, censée traiter et valider le formulaire.

Structure du formulaire

ex :
champs 'prénom',
champs 'email'

code HTML

je clique
sur submit

validation de formulaire

ex :
- est-ce que tous les champs ont bien été remplis ?
- est-ce que l'adresse email a une syntaxe correcte ?

code PHP

Indiquer au formulaire où se trouve la page de validation

```
<form action='validation.php'>
```

```
<!-- my form code here -->
```

```
</form>
```

Deux méthodes d'envoi

- Le protocole HTTP est celui qui permet la communication entre le client et le serveur. Il fonctionne par requête <-> réponse

Ex : le client envoie le formulaire au serveur via une requête HTTP, le serveur traite le formulaire et lui renvoie une réponse (par ex : 'ton email n'est pas correct')

- Les deux méthodes les plus communes sont **GET** et **POST**.
- Une différence notable est qu'avec GET les valeurs qu'on envoie apparaissent dans l'URL, par exemple : validation.php?[prenom=Bob&email=bob@site.com](#)
- Il faut indiquer la méthode d'envoi dans le code du formulaire.

Indiquer au formulaire la méthode d'envoi

```
<form action='validation.php' method='post'>
```

```
<!-- my form code here -->
```

```
</form>
```

Décrire ses champs avec <input>

- L'élément le plus courant pour les formulaires.

```
<input type='text' name='prenom' value='ton prenom'>
```

attribut **type** : le type de champs : texte, mot de passe, email, case à cocher..

attribut **name** : (**obligatoire**) le nom du champs, nécessaire pour la validation ensuite

attribut **value** : valeur par défaut qui s'affiche dans le champs

Différents types de <input> - Exemples

Texte

`<input type='text' .. >`

Valeur

Mot de passe

`<input type='password' .. >`

.....

Case à cocher

`<input type='password' .. >`

☐

Submit/Validation

`<input type='submit' .. >`

Valider



Challenges



<(^)>

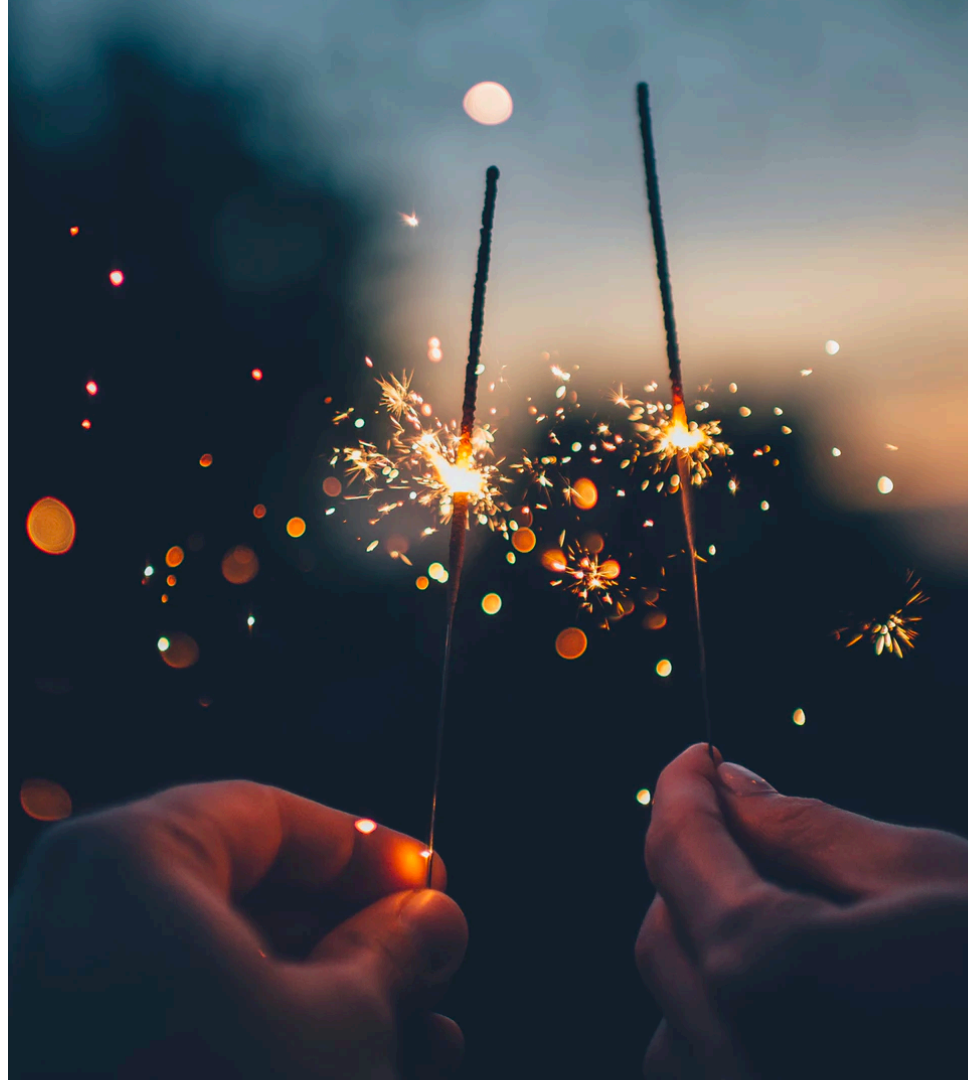
<{^}>

Travail en autonomie

</^>

<{^}>

Interactions utilisateurs en CSS



Interactions utilisateurs

- CSS permet de réaliser des effets suite à des interactions utilisateur.
- Exemple :
<http://2016.ouisharefest.com/>
 - > effet du halo de lumière, permet d'engager plus le visiteur
 - > effet sur les boutons au survol, améliorer l'expérience utilisateur
- Ces effets -même discrets- sont communs maintenant sur les site web et il est important de les connaître en CSS.
- Quand on pense à « animation », on pense souvent à JS, alors que de nombreuses choses sont déjà possibles en CSS !

Interactions animées avec CSS

Quelques exemples avant de passer à la pratique ;)

Changer le style au survol de la souris

<https://codepen.io/anon/pen/boLGyP?editors=1100>

Les transitions en CSS

Permettent d'introduire une transition lors d'un changement de valeur CSS

`<(^)>`

`<{^}>`

<https://codepen.io/anon/pen/RLQNaE?editors=1100>

Les transform en CSS

Permettent de bouger / animer des éléments de manière poussée

<(^)>

<{^}>

<https://codepen.io/anon/pen/QqQwKy?editors=1100>

</^>

<{^}>



Challenges



<(^)>

<{^}>

Travail en autonomie

</^>

<{^}>