

Recherche

Concept

Recherche

On souhaiterait :

- Afficher un champ de recherche et utiliser sa valeur
- Garder une URL dédiée à la recherche
- Afficher les résultats de recherche, comme une liste

Concept

Recherche



Comment intégrer la fonction de recherche ?

Concept

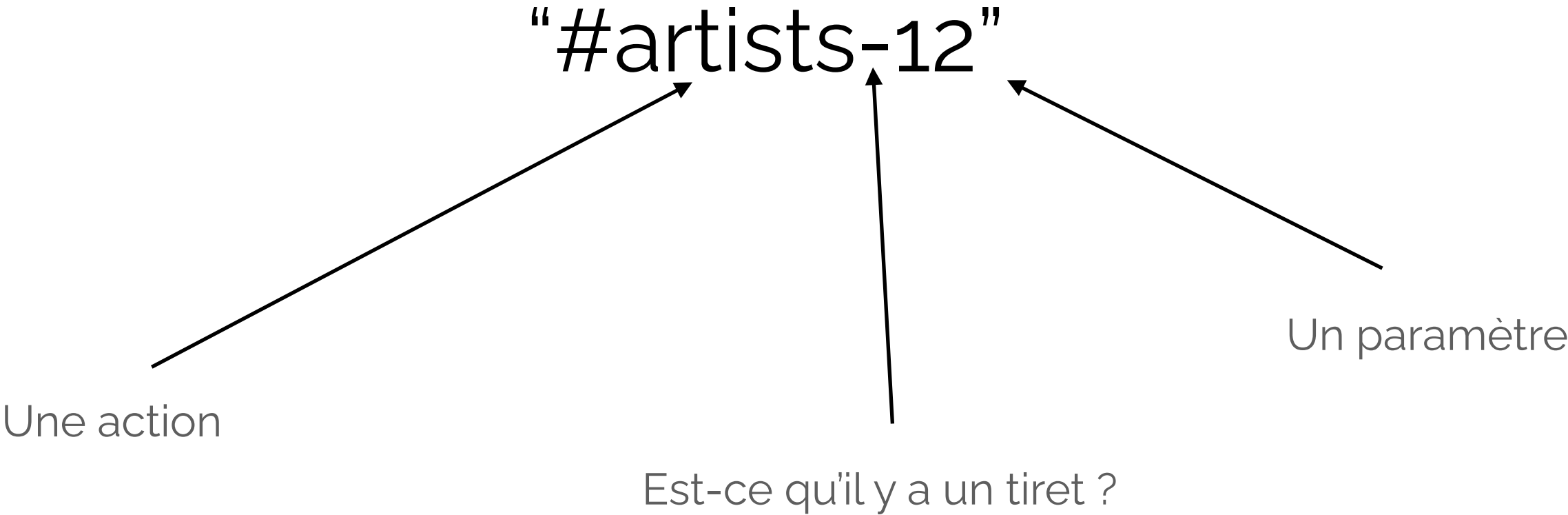
Recherche

“#artists-12”

Est-ce qu'il y a un tiret ?

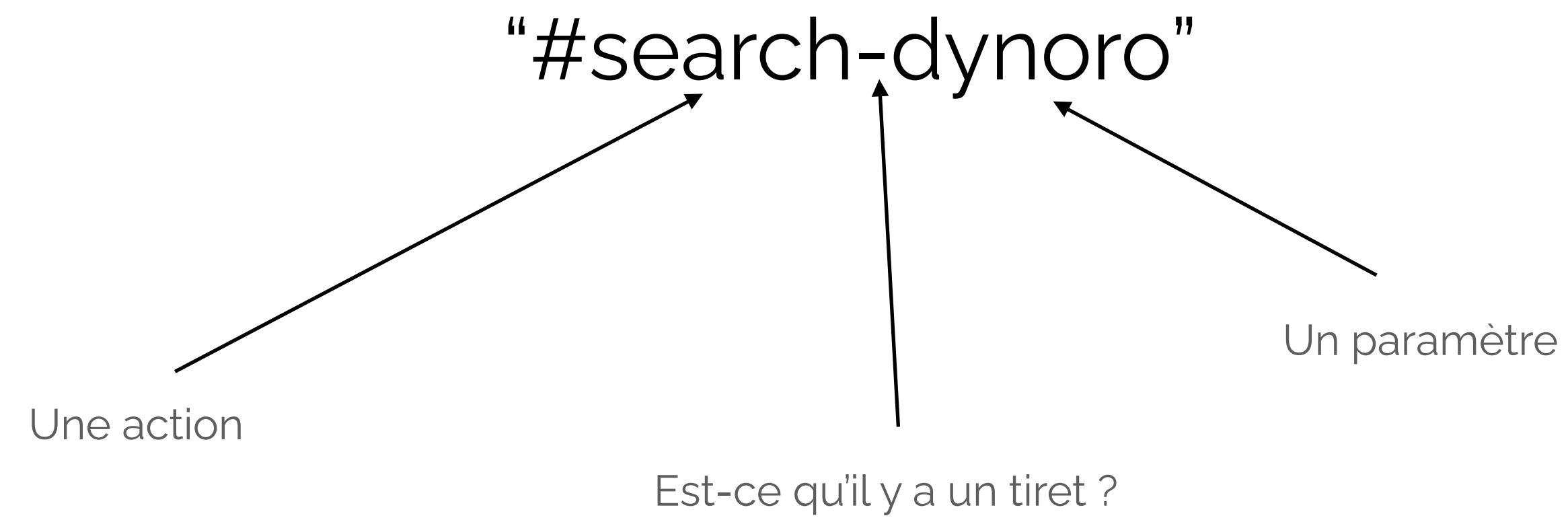
Concept

Recherche



Concept

Recherche



Concept

Recherche



Hint

`window.location.hash` est éditable

Structure

Recherche

- Créer un listener 'change' sur le champ de recherche
- Quand il change, il faut prendre sa valeur et la passer à l'endpoint de recherche
- La valeur se passe dans l'URL: /api/songs/search/:query
- Exemple: /api/songs/search/dynoro

Structure

Recherche

- Cela fonctionne bien pour de petites chaines, mais comment passer des caractères complexes ?
- `encodeURIComponent(laChaine)`
- Il faudrait donc faire:
- `fetch("https://api/songs/search/" + encodeURIComponent(laChaine))`

Structure

Recherche

- L'API de recherche vous renvoie la même structure que l'endpoint des chansons d'un artiste
- Vous pouvez donc réutilisez une bonne partie de votre code pour afficher résultat!
- Une fois le résultat obtenu, il vous suffira d'afficher la section correspondante pour la liste des résultats (de l'intérêt de bouger la logique d'affichage dans des petites méthodes d'aide :))

Favoris

Concept

Favoris

- Le but est de gérer les favoris localement
- En opposition à l'API qui gère les chansons de manière centralisée sur le serveur, nous utiliserons une des API Web pour les stocker dans le browser
- Plusieurs API disponibles...

Les API de stockage

Favoris

APIs de stockage :

- IndexedDB - Une vraie database, disponible dans le browser
- Web Storage - Stockage de clés/valeurs (genre de cookies sous stéroïdes)

Nous allons nous concentrer sur la deuxième...

L'API Web Storage

Favoris

Web Storage est séparée en 2 sous-API :

- **sessionStorage**
Les données ne sont stockées que pendant la durée de la session.
Lorsque le browser se ferme, les data sont effacées
- **localStorage**
Les données sont persistées entre les sessions et sont présentes après un restart

Ces deux APIs s'utilisent de la même manière, c'est uniquement la persistance des données qui change

L'API Web Storage

Favoris

Méthodes principales :

- `setItem(clé, valeur)` - Définit un élément "clé" avec comme valeur "valeur"
- `getItem(clé)` - Lit l'élément "clé"
- `removeItem(clé)` - Supprime l'élément "clé"
- `clear()` - Vide la totalité du localStorage

L'API Web Storage

Favoris

Exemple :

```
const uneValeur = 12345  
  
localStorage.setItem( 'uneCléAChoix', uneValeur )  
  
console.log( localStorage.getItem( 'uneCléAChoix' ) )  
  
==> "12345"
```


L'API Web Storage

Favoris

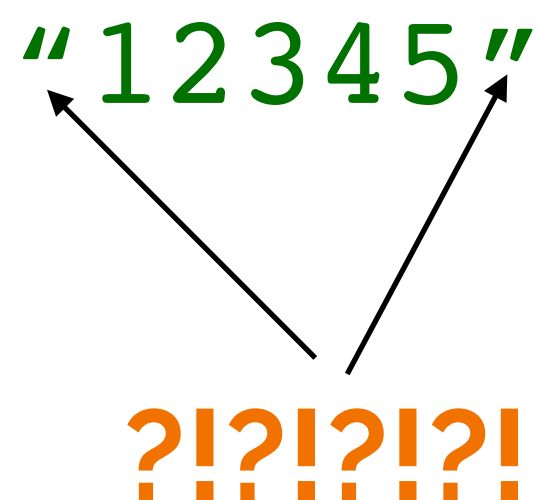
Exemple :

```
const uneValeur = 12345
```

```
localStorage.setItem( 'uneCléAChoix', uneValeur )
```

```
console.log( localStorage.getItem( 'uneCléAChoix' ) )
```

==> "12345"



?!?!?!?

localStorage ne stock que des chaînes de caractères...

L'API Web Storage

Favoris

- Est-ce un problème ?
- Oui et non... nous connaissons... JSON !
- Il est donc possible de créer des petites fonctions d'aide

```
const setItem = (id, value) => localStorage.setItem(id, JSON.stringify(value))
```

```
const getItem = (id) => JSON.parse(localStorage.getItem(id))
```

```
const getItems = () => Object.keys(localStorage).map(getItem)
```

```
const removeItem = (id) => localStorage.removeItem(id)
```

Trouver un favori

Favoris

Vérifier si une chanson est présente dans les favoris

```
if (getItem(idRecherché)) {  
  
    console.log("C'est dedans!")  
  
    // Par exemple, removeItem(idRecherché) pour la sortir des favoris  
  
} else {  
  
    console.log("ça n'y est pas...")  
  
    // Par exemple, setItem(idRecherché, song) pour l'ajouter aux favoris  
  
}
```

Structure

Favoris



Comment structurer le code ?

Structure

Favoris

On peut subdiviser le problème en plusieurs sous-problèmes :

- Ajouter/supprimer un élément de la liste des favoris (click sur le bouton ajout)
- Vérifier si un élément de la liste est présent (afficher le bon icône)
- Utiliser la liste des favoris pour afficher les chansons

1) Ajouter/supprimer un élément de la liste des favoris

Favoris

Il nous faut :

- Listener 'click' sur le bouton
- Une méthode pour ajouter/supprimer

2) Vérifier si un élément de la liste est présent

Favoris

Il nous faut :

- Une méthode pour tester si oui/non
- Un moyen de modifier le HTML facilement
- Hint: Slides cours 2, page 58

2) Vérifier si un élément de la liste est présent

Favoris

Hint bis: Les attributs sont des chaines.. quand on passe `null`, la valeur récupérée par `getAttribute` ne donne pas `null`, mais la chaine `"null"`

Par exemple :

```
if( this.getAttribute( "favorite" ) == "null" )
```

...

3) Utiliser la liste des favoris pour afficher les chansons

Favoris

Il nous faut :

- Un moyen de récupérer une liste de chansons
- Un moyen d'afficher une liste de chansons

Composants ?

Projet

- ✓ Projet parcel vide
- ✓ Squelette HTML
- ✓ Styles CSS structurels
- ✓ Icônes
- ✓ Routeur pour les pages web
- ✓ Client pour l'API JSON

- ✓ Lecteur audio
- ✓ Local storage pour les favoris
 - Détection online/offline
 - Manifest PWA
 - Caching
 - Service worker