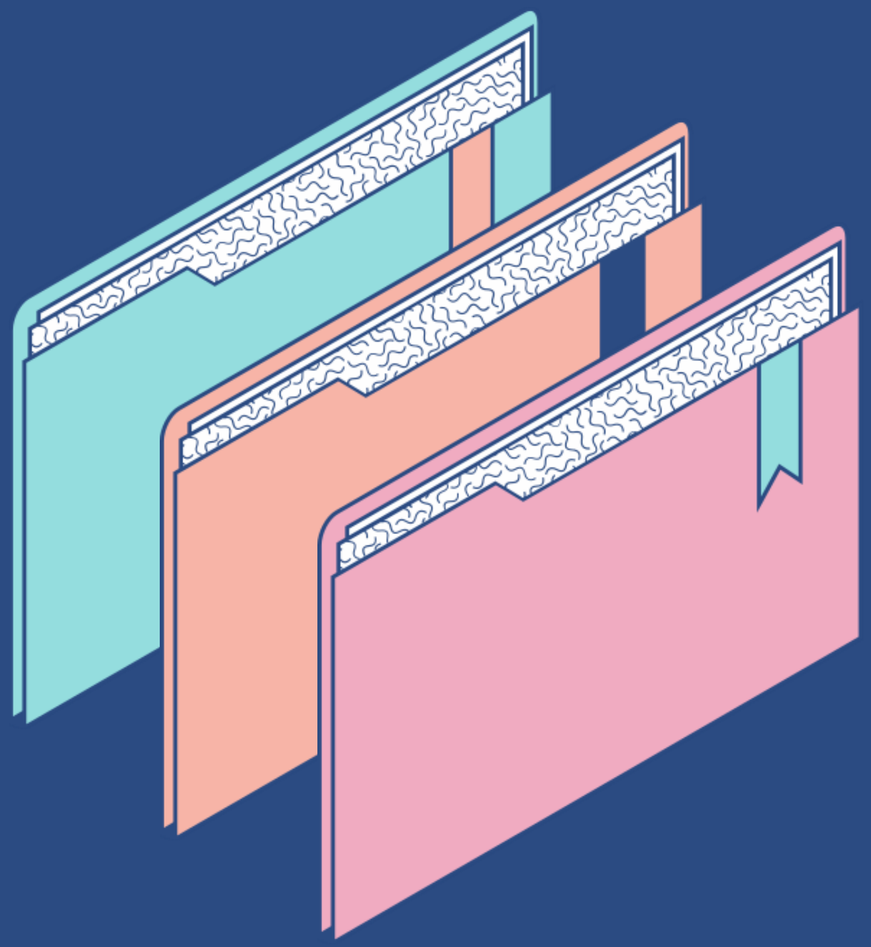
SOMMAIRE



Fonctions avancées


```
() ⇒ { expressions }
```

ES6 Arrow Functions

Arguments variables fonctions

Il est possible de définir un nombre variable d'arguments en entrée de fonction.
Cela s'effectue en ajoutant "..." avant l'argument, qui sera ensuite considéré comme un tableau au sein de la fonciton

```
let nom = 'Giraud', prenom = 'Pierre';
function profil(nom, prenom, ...hobbies){
    let h = '';
    for(hobbie of hobbies){
        h += hobbie + ', ';
    }
    alert('Nom : ' + nom + '\nPrénom : ' + prenom + '\nHobbies : ' + h);
}

/*Décommentez le code pour afficher les boites d'alerte
profil('Giraud', 'Pierre');
profil('Giraud', 'Pierre', 'Trail');
profil('Giraud', 'Pierre', 'Trail', 'Triathlon');
*/
```

Fonctions fléchées

Les fonctions fléchées possèdent une syntaxe très compacte.

```
let somme = function(a, b) {
    return a + b;
};
*/

//Equivalent en fonction fléchée :
let somme = (a, b) => a + b;
```

Timers

setTimeout() : Permet d'exécuter une fonction ou un bloc de code après une période définie.

```
let b1 = document.getElementById('b1');

b1.addEventListener('click', message);
function message(){
    setTimeout(alert, 2000, 'Message
d\'alerte après 2 secondes');
}
```

Timers

clearTimeout() :
Permet d'annuler ou supprimer un délai défini avec setTimeout().

```
let b1 = document.getElementById('b1');
let b2 = document.getElementById('b2');
let timeoutId;

b1.addEventListener('click', message);
b2.addEventListener('click', stopDelai);

function message(){
    timeoutId = setTimeout(alert, 2000,
'Message d\'alerte après 2 secondes');
}
function stopDelai(){
    clearTimeout(timeoutId);
}
```

Timers

setInterval() :

Permet d'exécuter une fonction ou bloc de code selon un intervalle de temps fixé.

```
let b1 = document.getElementById('b1');
let b2 = document.getElementById('b2');
let b3 = document.getElementById('b3');
let p1 = document.getElementById('p1');
let timeoutId;

b1.addEventListener('click', message);
b2.addEventListener('click', stopDelai);
b3.addEventListener('click', afficheHeure);

function message(){
    timeoutId = setTimeout(alert, 2000,
'Message d\'alerte après 2 secondes');
}
function stopDelai(){
    clearTimeout(timeoutId);
}

function afficheHeure(){
    setInterval(function(){
        let d = new Date();
        p1.innerHTML =
d.toLocaleTimeString();
    }, 1000)
}
```

Timers

clearInterval() :
Permet d'annuler l'exécution en boucle d'une fonction définie par setInterval()

```
}
function stopDelai(){
    clearTimeout(timeoutId);
}
```