

Gestion des erreurs

Try ... Catch

Aspect très important de la programmation d'autant plus quand on travaille dans une équipe, progressivement vos applications vont se complexifier et contenir beaucoup de code, d'objet, de fonctions etc..

Il y a des erreurs plus ou moins grave et de sources différentes

Les classiques :

- Erreur de Syntaxe, oubli d'un « ; », length ou lenght ?, etc...
- Erreur qui vient du serveur (pratique on peut accuser les développeurs BackEnd, l'incapacité à charger un fichier (404, etc...))
- Erreur qui vient du navigateur
- Erreur qui vient de l'utilisateur (envoyer une valeur non conforme dans un formulaire par exemple)

Dans la majorité des cas, une erreur va provoquer l'arrêt brutal d'un script et on risque donc d'avoir des codes et des pages non fonctionnelles.

Dans le pire des cas, une erreur peut être utilisée comme faille de sécurité par des utilisateurs malveillants qui vont l'utiliser pour dérober des informations ou pour tromper les autres visiteurs.

Javascript comporte nativement des fonctionnalités pour gérer les cas d'erreurs.

Dans tous les cas vous avez pu le constater, quand votre code comporte une erreur (vous avez un message d'erreur de base dans la console du navigateur), cela signifie que de base Javascript va créer, générer une erreur à partir de l'objet global Error (un objet de base dans javascript)

Par la suite nous allons pouvoir capturer l'objet d'Error renvoyé par Javascript et pouvoir indiquer ce que l'on souhaite faire dans le code quand cette erreur survient.

On va avoir à disposition un syntaxe de bloc try ... catch...

Dans le bloc try : on écrit le code à tester (qui peut potentiellement générer des erreurs)

Dans le bloc catch : on écrit le code pour gérer l'erreur (on fait un simple console.log ? On affiche un message à l'utilisateur ? On enregistre quelque chose en base de données ?

Nous allons voir le gestion d'erreurs au travers d'exemples.

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

```
prenom;  
alert('Ce message ne s\'affichera pas');
```

Erreur générée par la fonction alerte

✖ **Uncaught SyntaxError: Invalid or unexpected token (at app.js:4:7)** app.js:4

Erreur générée par la variable prénom

✖ ▶ **Uncaught ReferenceError: prenom is not defined** app.js:3
 at app.js:3:1

Donc si on sait que ce code d'exemple peut nous créer des erreurs on peut l'écrire de cette manière :

```
try{  
  prenom  
  alert('Bonjour');  
}catch(err){  
  alert(`Erreur Détectée ALERTE STOPPEZ TOUT :  
  -----  
  Le Nom de l'erreur  
  ${err.name}  
  -----  
  Le Message de l'erreur :  
  ${err.message}  
  -----  
  L'emplacement de l'erreur:  
  ${err.stack}`);  
}  
alert(`Ce message s'affiche correctement`);
```

Comme on a un erreur dans le code du bloc Try seule les alertes du bloc catch puis ensuite la dernière alert sera affichée à l'utilisateur.

Auteur :

Jean-François Pech

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
 ☑ Sophie POULAKOS
 ☑ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.



127.0.0.1:5500 says

Erreur Détectée ALERTE STOPPEZ TOUT:

Le Nom de l'erreur
ReferenceError

Le Message de l'erreur :
prenom is not defined

L'emplacement de l'erreur:
ReferenceError: prenom is not defined
at http://127.0.0.1:5500/app.js:7:5

☐ Don't allow this page to display more dialogs

OK



127.0.0.1:5500 says

Ce message s'affiche correctement

☐ Don't allow this page to display more dialogs

OK

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023

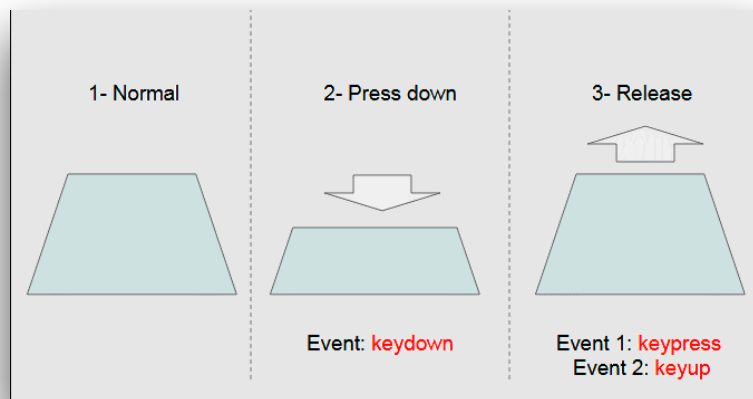


Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Exercice : réagir au clavier

Dans ce cas là plusieurs évènements s'offrent à nous, keyup, keypress ou keydown, cela correspond aux différentes phases lorsque l'on presse une touche du clavier.

Plus explicite :



Généralement pour pouvoir exploiter ce que tape l'utilisateur on va privilégier l'évènement keyup. (Keydown rarement (les long jump dans sonic 2 🎮 ?))

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Dans une page web : rajouter un formulaire (au moins un textarea)

Dans votre programme JS :

No Spoil :)

D.O.M Events

Email address

Title


Example textarea

azertyuiop


Submit

[Ajouter Classe](#)
[Suppr Classe](#)
[Toggle Classe](#)


S’AFFICHE ICI ➡ azertyuiop




Card title



Card title



Card title



Card title

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Solution Possible

```
const monTxt = document.querySelector("textarea");
const rendu = document.querySelector("#formRender");

monTxt.addEventListener("keyup", function() {
    rendu.innerHTML = monTxt.value;
});
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Exercice : Réagir au clavier 2

Dans une page web on a un input ou textarea, et un bouton, faire en sorte que :

Au bout de 5 lettres tapées au clavier cela Désactive le bouton :

Example textarea

rrrrr|

Submit

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023






Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Web Storage

Dans le navigateur on peut profiter de plusieurs espaces pour pouvoir sauvegarder des données, vous connaissez déjà (surtout depuis RGPD) les cookies qui permettent par exemple d'enregistrer un email et mot de passe dans un formulaire, ou reprendre l'épisode d'une série au moment où nous l'avions quitté, etc ...

Ce schéma récapitule les différentes caractéristiques de ces principaux espaces de stockage.

			
Criteria	Local Storage	Session Storage	Cookies
Storage Capacity	5-10 mb	5-10 mb	4 kb
Auto Expiry	No	Yes	Yes
Server Side Accessibility	No	No	Yes
Data Transfer HTTP Request	No	No	Yes
Data Persistence	Till manually deleted	Till browser tab is closed	As per expiry TTL set

Dans l'exemple suivant nous allons nous intéresser au local Storage (la façon de l'utiliser est assez similaire pour la session et les cookies)

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

⚠ Gardez en tête que ces espaces de stockage ne sont pas définitifs, l'utilisateur peut toujours aller dans les paramètres du navigateur (ou si il est navigation privée) et supprimer ces espaces de stockage, de plus ils ont une capacité de stockage limitée.

Local Storage, évidemment il y a un objet localStorage avec des fonctions natives de JS permettant de gérer cet espace de mémoire commun à tous les navigateurs.

La fonction setItem()

qui prend 2 paramètres, (une clé et une valeur), ci dessous dans le local Storage on sauvegarde une clé 'lastname' avec une valeur à 'smith'

```
localStorage.setItem("lastname", "Smith");
```

La fonction getItem()

qui prend 1 paramètres, (la clé), ci dessous dans le local Storage on récupère le contenu de la clé 'lastname'

```
localStorage.getItem("lastname");
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023

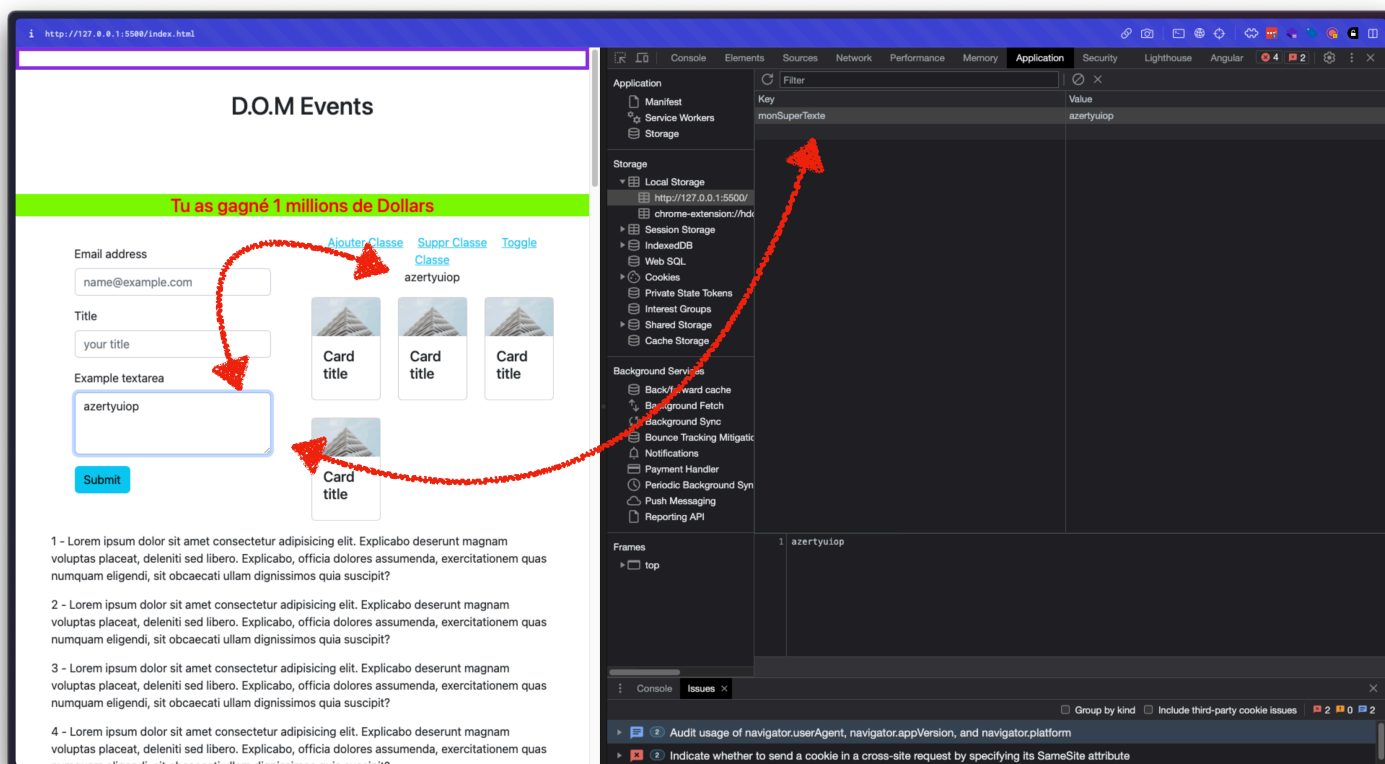


Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Exercice : LocalStorage

En se basant sur l'exercice précédent (le mini éditeur de texte)

Dans une page web avec un formulaire (ou au moins un textarea)



Dans votre programme JS :

- Stocker l'input ou textarea dans une variable monTxt
- Stocker la div (ou le titre ou paragraphe etc..) dans une variable rendu
- À la valeur du textarea assigner ce qu'on récupère dans le local Storage sous le nom (la clé) « monSuperTexte »
- Ensuite mettre en place une condition : Si la valeur de monTxt est définie ALORS ,le innerText du rendu récupère ce qu'il y a dans le local Storage sous le nom (la clé) « monSuperTexte »
- Après cette condition mettre en place une écoute de l'événement du clavier sur monTxt et faire :
 - Placer dans le local Storage la valeur de monTxt sous le nom « monSuperTexte »
 - Afficher la valeur de monTxt dans le rendu

En console dans l'onglet Application :

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

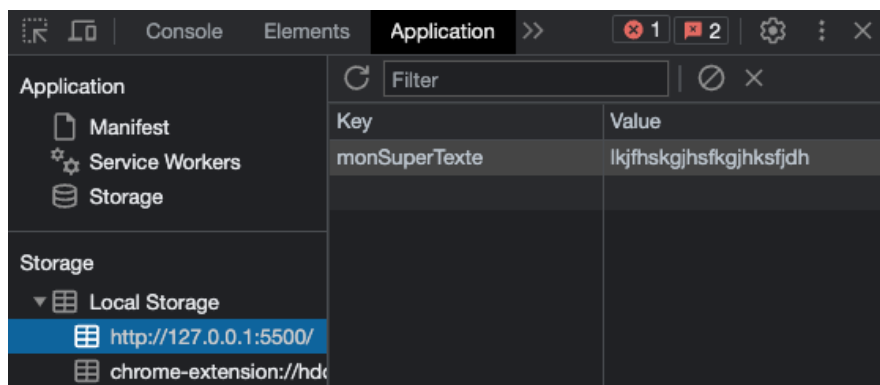
☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.



Quand on recharge la page ou quitter et relancer le navigateur le mini éditeur de texte est pré rempli et son affichage aussi

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

Solution Possible

```
const monTxt = document.querySelector("#formMessage");
const rendu = document.querySelector("#formRender");
// On va pré remplir le textarea avec ce que l'on récupère dans le localStorage
monTxt.value = localStorage.getItem('monSuperTexte');
//Si monTxt.value est défini alors on remplit la Div avec ce qu'on récupère dans le
local storage
if(monTxt.value){
    rendu.innerHTML = localStorage.getItem('monSuperTexte');
};
//On détecte ce que tape l'utilisateur dans le textarea
monTxt.addEventListener("keyup", function() {
//On enregistre ce que tape l'utilisateur dans le localStorage sous le nom
"monSuperTexte"
localStorage.setItem('monSuperTexte',monTxt.value);
//On affiche ce que tape l'utilisateur dans la div
    rendu.innerHTML =monTxt.value;
});
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.