

# CRÉEZ UNE LANDING PAGE AVEC JAVASCRIPT

Yixuan WANG

mercredi 1 décembre 2021, 13:00



# L'idée du projet & Le découpage des tâches

Validation de formulaire côté client

1

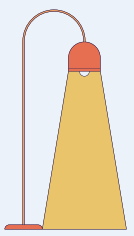
Lancer, soumettre et fermer le formulaire

2

Envoyer un message d'erreur ou de réussite en fonction de la valeur saisie dans `<input>`

3

Envoyer un message de remerciement lorsque toutes les valeurs d'entrée sont validées



# Lancer, soumettre et fermer le formulaire

## Lancer :

```
const launchModal = function () {  
  modalbg.style.display = "block";  
};
```

Sign Up Now

```
signUpBtn.addEventListener("click", (event) => {  
  event.preventDefault();  
  launchModal();  
});
```



event.preventDefault();  
=> if the event does not get explicitly handled, its default action should not be taken as it normally would be.

## Soumettre :

Go

```
form.addEventListener("submit", (event) => {  
  event.preventDefault();  
  isInput();  
});
```

Add 1st event to submit-button for submit:

```
const addSubmitEvent = function (event) {  
  event.preventDefault();  
  onSubmit();  
  isInput();  
};  
submitButton.addEventListener("click", addSubmitEvent, false);
```

## Fermer :

```
const closeModal = function () {  
  modalbg.style.display = "none";  
};
```



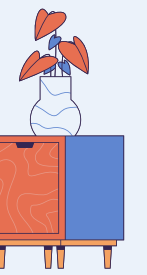
```
closeBtn.addEventListener("click", (event) => {  
  event.preventDefault();  
  closeModal();  
});
```

Add 2nd event to submit-button for close:

Close

```
submitButton.innerHTML = submitButton.innerHTML.replace("Go", "Close");  
const addCloseEvent = function (event) {  
  event.preventDefault();  
  closeModal();  
};  
submitButton.addEventListener("click", addCloseEvent, false);
```

# Envoyer un message d'erreur ou de réussite



```
function isError(){...}  
function isSuccess(){...}
```



```
// Setup 2 functions = isError() & isSuccess():  
function isError(input, message) {  
  const formData = input.parentElement;  
  const small = formData.querySelector("small");  
  //add error message inside small element  
  small.innerHTML = message;  
  //add error class  
  formData.className = "formData error";  
}  
function isSuccess(input) {  
  const formData = input.parentElement;  
  //add success class  
  formData.className = "formData success";  
}
```

Tuto source : <https://www.youtube.com/watch?v=rsd4FNGTRBw>

# function isInput(){...}

First name

g

❗ Please enter 2 or more characters for the first name field

```
if (firstNameValue.length < 2) {  
  // show error message  
  isError(  
    firstName,  
    "Please enter 2 or more characters for the first name field"  
  );  
} else {  
  // show success green icon  
  isSuccess(firstName);  
}
```

```
if (birthdayValue === "") {  
  // show error message  
  isError(birthday, "Birthday cannot be empty");  
} else {  
  // show success green icon  
  isSuccess(birthday);  
}
```

Last name

de



```
if (lastNameValue.length < 2) {  
  // show error message  
  isError(  
    lastName,  
    "Please enter 2 or more characters for the last name field"  
  );  
} else {  
  // show success green icon  
  isSuccess(lastName);  
}
```

```
if (quantityValue === "") {  
  // show error message  
  isError(quantity, "Tournament cannot be empty");  
} else {  
  // show success green icon  
  isSuccess(quantity);  
}
```

# function isInput(){...}

Email

❗ Email cannot be empty

Email

❗ Please provide a valid email address

Email

✅

```
const emailFormat = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;
```

```
//Email test:  
function isEmail(email) {  
  return emailFormat.test(email);  
}
```

```
if (emailValue === "") {  
  // show error message  
  isError(email, "Email cannot be empty");  
} else if (!isEmail(emailValue)) {  
  isError(email, "Please provide a valid email address");  
} else {  
  // show success green icon  
  isSuccess(email);  
}
```

# function isInput(){...}

Which location?

☐ New York   ☐ San Francisco   ☐ Seattle   ☐ Chicago  
☐ Boston   ☐ Portland

❗ You have to choose a city

☐ I have read and accept the terms of use.

☐ I want to be notified of upcoming events. (Optional)

❗ You have to accept the terms and conditions

Which location?

☒ New York   ☐ San Francisco   ☐ Seattle   ☐ Chicago  
☐ Boston   ☐ Portland



☒ I have read and accept the terms of use.

☐ I want to be notified of upcoming events. (Optional)



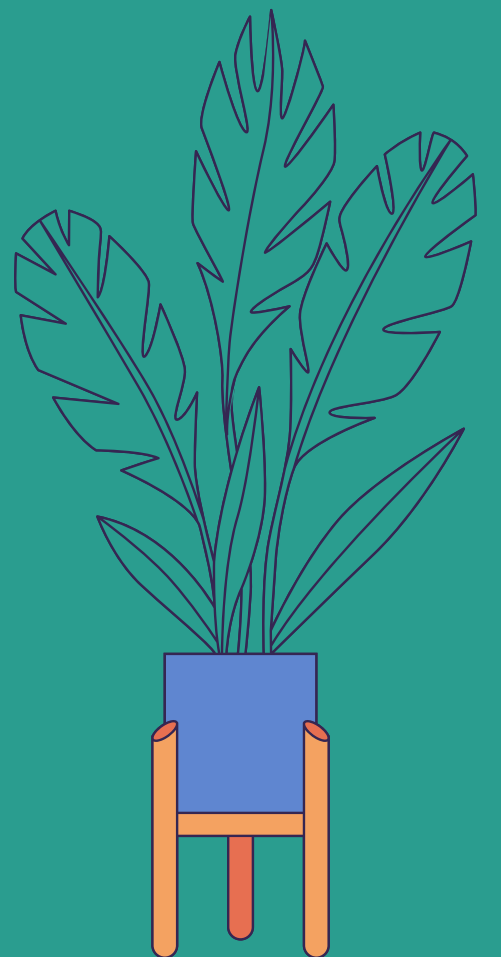
```
//for Radio checkbox part :  
for (const location of locations) {  
  if (location.checked) {  
    isSuccess(locationMessage);  
    break;  
  } else {  
    isError(locationMessage, "You have to choose a city");  
  }  
}
```

```
//for checkbox part :  
if (checkbox1.checked) {  
  isSuccess(conditionMessage);  
} else {  
  isError(conditionMessage, "You have to accept the terms and conditions");  
}
```

# Valider le formulaire & Envoyer un message de réussite

```
if (
  firstNameValue.length > 1 &&
  lastNameValue.length > 1 &&
  emailValue.match(emailFormat) &&
  birthdayValue !== "" &&
  quantityValue !== "" &&
  checkbox1.checked
) {
  form.remove();
  thanksMessage.style.display = "block";
  submitButton.innerHTML = submitButton.innerHTML.replace("Go", "Close");
  // add 1st event to submit-button:
  const addCloseEvent = function (event) {
    event.preventDefault();
    closeModal();
  };
  submitButton.addEventListener("click", addCloseEvent, false);
} else {
  form.style.display = "block";
}
```

Si (on remplit tous les conditions){  
(1). faire disparaître le formulaire;  
(2). faire paraître le message "thanks";  
(3). changer le bouton "Go" en "Close";  
} sinon {  
Garder le formulaire;  
}





# Conclusion

## LES NOTIONS APPRIS

Variable

Condition

Function

DOM

Test manuel : `(console.log("..."))`

## BUG

Which location?

☐ New York ☐ San Francisco ☐ Seattle ☐ Chicago

☐ Boston ☐ Portland

Thank you for  
submitting your  
registration details!

Close

```
firstNameValue.length > 1 &&  
lastNameValue.length > 1 &&  
emailValue.match(emailFormat) &&  
birthdayValue !== "" &&  
quantityValue !== "" &&  
checkbox1.checked
```