FORM VALIDATION

1/ Function Validator:

```
function Validator(options) {
  let form = document.querySelector(options.form);
  let selectorRules = {};

  function getFormGroup(formInput) {
    while (formInput && !formInput.matches(options.formGroup))
      formInput = formInput.parentElement;
    return formInput;
  }

  function switchStatus(status, formInput, message) {
    let formGroup = getFormGroup(formInput);
    let formMessage = formGroup.querySelector(options.formMessage);

    switch (status) {
      case 'valid':
        formGroup.classList.add('valid');
        formGroup.classList.remove('invalid');
        formMessage.innerText = options.successMessage || '';
        break;
      case 'invalid':
        formGroup.classList.add('invalid');
        formGroup.classList.remove('valid');
        formMessage.innerText = message || 'An error has occur';
        break;
      case 'neutral':
        formGroup.classList.remove('invalid');
        formGroup.classList.remove('valid');
        formMessage.innerText = '';
        break;
      default:
        break;
    }
  }

  function validate(formInput, rule) {
    let rules = selectorRules[rule.selector];
    let errorMessage = '';

    for (let i = 0; i < rules.length; i++) {
      switch (formInput.type) {
        case 'checkbox':
        case 'radio':
          let checkedInput =
getFormGroup(formInput).querySelector(rule.selector + ':checked');
          errorMessage = rules[i](checkedInput && checkedInput.value);
          break;
        default:
          errorMessage = rules[i](formInput.value);
      }
      if (errorMessage) {
        switchStatus('invalid', formInput, errorMessage);
        break;
      }
      else
        switchStatus('valid', formInput);
    }
    return !errorMessage;
  }

  options.rules.forEach(rule => {
    if (!Array.isArray(selectorRules[rule.selector]))
      selectorRules[rule.selector] = [rule.test];
    else
      selectorRules[rule.selector].push(rule.test);

    let formInputs = form.querySelectorAll(rule.selector);

    formInputs.forEach(formInput => {
      if (formInput) {
        formInput.onblur = () => {
          validate(formInput, rule);
        }
        if (options.successMessage === '' || formInput.type !==
'checkbox')
          formInput.oninput = () => {
            switchStatus('neutral', formInput);
          }
      }
    })
  })

  form.onsubmit = (e) => {
    let validForm = true;

    options.rules.forEach(rule => {
      let formInput = form.querySelector(rule.selector);

      if (formInput) {
        let validInput = validate(formInput, rule);
        if (!validInput)
          validForm = false;
      }

    })
    if (!validForm) {
      e.preventDefault();
    }
  }
}

Validator.isRequired = (selector, message) => ({
  selector: selector,
  test: (value) => value ? undefined : message || 'Error'
});
Validator.isEmail = (selector, message) => ({
  selector: selector,
  test: (value) => {
    let regex = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
    return regex.test(value) ? undefined : message || 'Error';
  }
});

// Email Regex:
// /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/
```

2/ Usage:

a/ In the HTML file containing the form

```
 <script src="./validation.js"></script>
  <script>
    Validator({
      form: '#form',
      formGroup: '.form-group',
      formMessage: '.form-message',
      successMessage: 'Passed',
      rules: [
        Validator.isRequired('#full-name', 'Your name is required'),
        Validator.isRequired('#email', 'Your email is required'),
        Validator.isEmail('#email', 'The email is invalid'),
        Validator.isRequired('input[type="checkbox"]', 'Please choose
at least a language'),
      ],
    });
  </script>
```

b/ For the HTML file containing the information (Query string)

```
let params = new URL(window.location).searchParams;
let fullName = params.get('full-name');
let email = params.get('email');
let languages = params.getAll('languages').join(', ');
let formResult = document.querySelector('.form-result');

formResult.innerHTML +=
  `
  <div class="form-result-group">
    <span class="form-result-key">Full name:</span>
    <span class="form-result-value">${fullName}</span>
  </div>
  <div class="form-result-group">
    <span class="form-result-key">Email:</span>
    <span class="form-result-value">${email}</span>
  </div>
  <div class="form-result-group">
    <span class="form-result-key">Chosen languages:</span>
    <span class="form-result-value">${languages}</span>
  </div>
  `;
```

HTML & CSS

1/ Sticky footer

a/ HTML

```
<div class="wrapper">
  <header class="header"></header>
  <main class="main"></main>
  <footer class="footer"></footer>
</div>
```

b/ CSS

```
.wrapper {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
  gap: 120px;
}
.footer {
  margin-top: auto;
}
```

2/ Colors:

```
 :root {
  --green: #20E3B2;
  --blue: #2CCCFF;
  --purple: #6A5AF9;
  --pink: #F62682;
  --orange: #FF6651;
}
```