HTML5

HTML 5

- Dernière version de HTML (28 oct. 2014)
- Rich Interface Applications
- Langage balisé
- Basé sur XHTML

HTML 5 & Adobe Flash

- Adobe Flash est un **complément** d'HTML5
- Taux de pénétration :
  - Encore faible (75 %) pour HTML5
  - **Flash** (≈ 99 % sur PC, ≈ 80 % sur mobile) **supérieur à JavaScript** (≈ 90 %) !

2011 →

HTML 5 simplification

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
    <head>
        <title>Titre de la page</title>
        <meta http-equiv="content-type" content="text/html;charset=utf-8" />
    </head>
    <body>

    <!-- Ici votre site Web -->

    </body>
</html>
```

HTML 5 simplification

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
    <head>
        <title>Titre de la page</title>
        <meta http-equiv="content-type" content="text/html;charset=utf-8" />
    </head>
    <body>

    <!-- Ici votre site Web -->

    </body>
</html>
```

HTML 5 simplification

```
<!DOCTYPE html >
<html lang="fr" >
    <head>
        <title>Titre de la page</title>
        <meta charset="utf-8" />
    </head>
    <body>

    <!-- Ici votre site Web -->

    </body>
</html>
```

HTML5 : CSS / JS

- XHTML

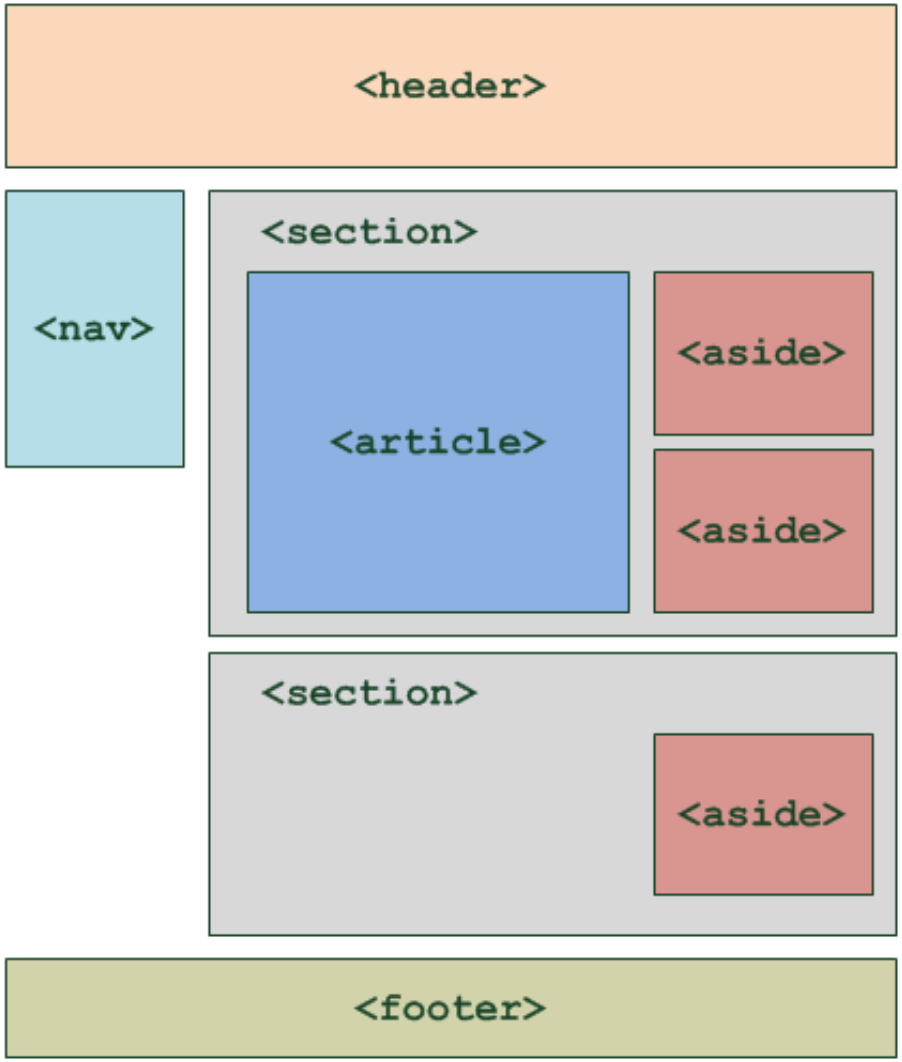
```
<head>
    <link rel="stylesheet" type="text/css"
href="css/monfichier.css" />
    …
    <script type="text/javascript" src="js/myFile.js" ></script>
</head>
```

- HTML5

```
<head>
    <link rel="stylesheet" href="css/monfichier.css" />
    …
    <script src="js/myFile.js" ></script>
</head>
```

Sémantique

- Entête
  `<header></header>`
- Pied de Page
  `<footer></footer>`
- Navigation
  `<nav></nav>`
- Zone de regroupement
  `<section></section>`
- Zone de contenu
  `<article></article>`
- Zone secondaire
  `<aside></aside>`



Multimédia

- Audio

```
<audio controls="controls">
    <source src="audio.ogg" type="audio/ogg" />
    <source src="audio.mp3" type="audio/mp3" />
    <source src="audio.wav" type="audio/wav" />
    Ici l'alternative: un lien de téléchargement, un message, etc.
</audio>
```

- Vidéo

```
<video controls="controls">
    <source src="video.ogv" type="video/ogg" />
    <source src="video.mp4" type="video/mp4" />
    Ici l'alternative : un lien de téléchargement, un message, etc.
</video>
```

Multimédia : codecs

HTML5 Audio Codecs

	MAC						WIN								
	CHROME	FIREFOX	OPERA		SAFARI		CHROME	FIREFOX	OPERA	IE					
	25	20	12	15	5.1	6	25	15	12	6	7	8	9	10	
Audio: ogg/vorbis	✔	✔	✔	✔	✘	✘	✔	✔	✔	✘	✘	✘	✘	✘	72%
Audio: mp3	✔	✘	✘	✘	✔	✔	✔	✘	✘	✘	✘	✘	✔	✔	50%
Audio: wav	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✘	✘	77%
Audio: AAC	✔	✘	✘	✘	✔	✔	✔	✘	✘	✘	✘	✘	✔	✔	50%

HTML5 Video Codecs

	MAC						WIN								
	CHROME	FIREFOX	OPERA		SAFARI		CHROME	FIREFOX	OPERA	IE					
	25	20	12	15	5.1	6	25	15	12	6	7	8	9	10	
Video: ogg/theora	✔	✔	✔	✔	✘	✘	✔	✔	✔	✘	✘	✘	✘	✘	71%
Video: H.264	✔	✘	✘	✘	✔	✔	✔	✘	✘	✘	✘	✘	✔	✔	50%
Video: WebM	✔	✔	✔	✔	✘	✘	✔	✔	✔	✘	✘	✘	✘	✘	58%

Images

- Image <img/>
- Regroupement <figure></figure>
- Légende <figcaption></figcaption>

```
<figure>
    <img src="/images/chaton.jpg" alt="Oh le beau chaton"/>
    <figcaption>Un petit chat mignon tout plein</figcaption>
</figure>
```

Conteneurs

- Division `<div></div>`
- Section `<section></section>`
- Article `<article></article>`
- Cadre `<iframe></iframe>`

Texte

- Titre `<h1></h1>` … `<h6></h6>`
- Force `<strong></strong>` Gras `<b></b>`
- Emphase `<em></em>` Italique `<i></i>`
- Surligné `<mark></mark>` Souligné `<u></u>`

Formulaires

- Conteneur `<form></form>`
  - Méthode d'envoi `method`
  - Encodage des données `enctype`
  - Cible d'exécution `action`
- Zone de texte `<textarea></textarea>`
- Liste de choix `<select></select>`, `<option></option>`
  - Liste à choix multiple `multiple="multiple"`
- Etiquette `<label for="id"></label>`

Formulaires

- Saisie d'information `<input/>`
  - Valeur initiale `value`
  - Nom de la donnée `name`
  - Désactivation `disabled`
  - Type de donnée `type`
- Valeurs possibles pour l'attribut type

image	password	text	url	date	month
reset	checkbox	hidden	number	time	week
button	file	tel	color	datetime	search
submit	radio	email	range	datetime-local	

image

reset

button

submit

password

checkbox

file

radio

text

hidden

tel

email

Image 

Reset Annuler

Button Bouton

Submit Appuyez

Password ••••••••

Checkbox

File Parcourir... Aucun fichier sélectionné.

Radio

Text Ici un texte

Hidden

Tel +33 6 12 34 56 78

Email contact@w3c.org

url

number

color

range

date

time

datetime

datetime-local

month

week

search

Url http://www.w3c.org

Number 42

Color

Range

Date

Datetime

Datetime-local

Month

Week

Search

Formulaires - Validation

- required
- placeholder
- pattern
- min / max / step
- list

- novalidate
- formnovalidate

Formulaires – Validation

checkValidity

setCustomValidity

validationMessage

willValidate

validity

validity.valid

validity.valueMissing

validity.customError

validity.typeMismatch

validity.rangeOverflow

validity.rangeUnderflow

validity.stepMismatch

validity.tooLong

validity.patternMismatch

XHR2

- XML HttpRequest 1 :

```
var xhr = new XMLHttpRequest();
xhr.open('GET', '/path/to/image.png', true);
xhr.overrideMimeType('text/plain; charset=x-user-defined');

xhr.onreadystatechange = function(e) {
    if (this.readyState == 4 && this.status == 200) {
        var binStr = this.responseText;
        for (var i = 0, len = binStr.length; i < len; ++i) {
            var c = binStr.charCodeAt(i);
            var byte = c & 0xff;
        }
    }
};

xhr.send();
```

XHR2

- XML HttpRequest 2 :

```
var xhr = new XMLHttpRequest();
xhr.open('GET', '/path/to/image.png', true);
xhr.responseType = 'blob';

xhr.onload = function(e) {
    if (this.status == 200) {
        // Note: .response instead of .responseText
        var blob = new Blob([this.response], {type: 'image/png'});
        ...
    }
};

xhr.send();
```

XHR2

- Types de réponses possibles
- text
- blob
- arrayBuffer

CORS

- Cross Origin Resource Sharing
- Accès à des ressources externes
- Indentification du partage

```
xhr.open('GET','http://datahost.example/data.xml');
```

Réponse (entête)		Requête (entête)
Access-Control-Max-Age	Access-Control-Allow-Origin	Origin
Access-Control-Expose-Headers	Access-Control-Allow-Headers	Access-Control-Request-Method
Access-Control-Allow-Credentials	Access-Control-Allow-Methods	Access-Control-Request-Headers

JSON

- JavaScript Object Notation
- Echange de données simplifié
- Indépendant du langage
- Tableaux imbriqués {} et []

```
{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

JSONP

- JSON with Padding
- Simplification des utilisations JavaScript
- Script Element Injection

```
traiterReponse{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

Messaging

- Communication avec iframe

```
var iframeWin = document.getElementById("cadre").contentWindow;
iframeWin.postMessage("Message", "http://www.lehtml.com");
```

```
if (window.addEventListener) {
    window.addEventListener("message", afficherMessage, false);
} else {
    window.attachEvent("onmessage", afficherMessage);
}
```

WebSockets

- Principe similaire à une socket
- Informations HTML5
- Utilisation JavaScript

```
var ws = new WebSocket("ws://echo.websocket.org");

ws.onopen = function(evt) { console.log("Connection open ..."); };
ws.onclose = function(evt) { console.log("Connection closed."); };
ws.onerror = function(evt) { console.log("WebSocket error : " + evt.data) };
ws.onmessage = function(evt) { console.log( "Received Message: " + evt.data); };
```

WebWorkers

- Equivalent d'un Thread en JavaScript
- WebWorker dédié

```
w = new Worker("SomeWorker.js");
w.onmessage = function(event) {
document.getElementById("result").innerHTML
        = event.data;
        };
w.terminate();
```

```
var i = 0;

function timedCount() {
    i = i + 1;
    postMessage(i);
    setTimeout("timedCount()",500);
}

timedCount();
```

WebWorkers

- WebWorker partagé

```
myWorker = new SharedWorker("SomeWorker.js");

myWorker.port.postMessage(value);
```

```
var port = e.ports[0];

port.postMessage(workerResult);
```

Geolocalisation

- Récupération de la position de l'utilisateur

```
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        document.getElementById("demo").innerHTML =
            "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    document.getElementById("demo").innerHTML =
        "Latitude: " + position.coords.latitude +
        "<br>Longitude: " + position.coords.longitude;
}
```

Geolocalisation

- Suivi de la position de l'utilisateur

```
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.watchPosition(showPosition);
    } else {
        document.getElementById("demo").innerHTML =
            "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    document.getElementById("demo").innerHTML =
        "Latitude: " + position.coords.latitude +
        "<br>Longitude: " + position.coords.longitude;
}
```

Geolocalisation

- Présentation sur une carte

```
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        document.getElementById("demo").innerHTML =
            "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
 var latlon = position.coords.latitude + "," + position.coords.longitude;

    var img_url = "http://maps.googleapis.com/maps/api/staticmap?center="
        + latlon + "&zoom=14&size=400x300&sensor=false";

    document.getElementById("mapholder").innerHTML = "<img src='" + img_url + "'>";
}
```

Geolocalisation

- Gestion des erreurs

```
navigator.geolocation.getCurrentPosition(showPosition, showError);
. . .
function showError(error) {
    var x = document.getElementById("demo");
    switch(error.code) {
        case error.PERMISSION_DENIED:
            x.innerHTML = "User denied the request for Geolocation."; break;
        case error.POSITION_UNAVAILABLE:
            x.innerHTML = "Location information is unavailable."; break;
        case error.TIMEOUT:
            x.innerHTML = "The request to get user location timed out."; break;
        case error.UNKNOWN_ERROR:
            x.innerHTML = "An unknown error occurred."; break;
    }
}
```

Mode Deconnecté

- Utilisation d'un cache manifest

```
<html manifest="cache.manifest">
. . .
```

```
CACHE MANIFEST

CACHE:
index.html
/favicon.ico

NETWORK:
*

FALLBACK:
images/online.png images/offline.png
```

Canvas

- Conteneur de dessin
- Fonctionnalité HTML5
- Remplissage par JavaScript

```
<canvas id="c1" width="200" height="100" style="border:1px solid #000000;" ></canvas>
```

Canvas - Dessin

- Début de dessin : `beginPath ()`
- Fin de dessin : `closePath ()`
- Positionnement du curseur : `moveTo ()`
- Tracé d'un trait : `stroke ()`
- Remplissage d'une forme : `fill ()`
- Ecriture de texte : `fillText()` / `strokeText()`

Canvas - Lignes

- Dessin d'une ligne : `lineTo()`
- Largeur d'une ligne : `lineWidth`
- Couleur d'une ligne : `strokeStyle`
- Fin de ligne : `lineCap` (`butt`, `round`, `square`)
- Jonction de lignes : `lineJoin`

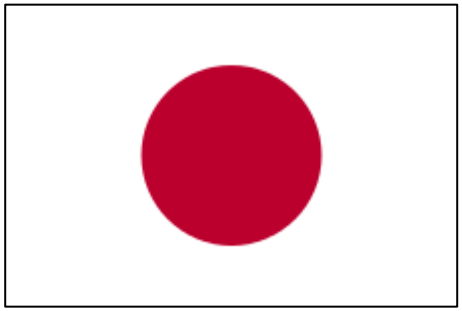
Canvas - Formes

- Dessin d'un arc : `arc()` / `arcTo ()`
- Courbe Quadratique : `quadraticCurveTo ()`
- Courbe de Bézier : `bezierCurveTo ()`
- Rectangle : `rect ()`

Canvas - Remplissage

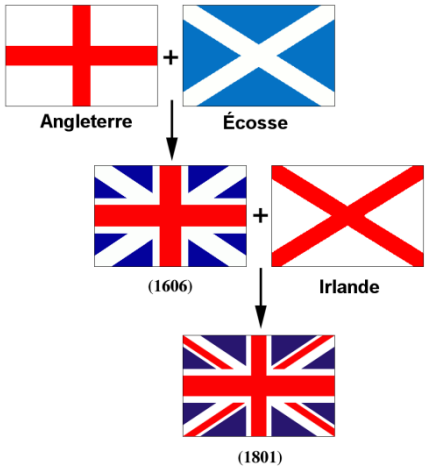
- Remplissage : `fill ()`
- Couleur de Remplissage : `fillStyle`
- Gradient Linéaire : `createLinearGradient ()`
- Gradient Circulaire : `createRadialGradient ()`
- Patron : `pattern ()`

Canvas – Exercice 1



Canvas – Exercice 2

Canvas – Exercice 3



Canvas - Texte

- Ecriture de texte (plein) : `fillText()`
- Ecriture de texte (contour) : `strokeText()`
- Police de caractères : `font`
  - Contient police, taille et style
- Alignement (horizontal) : `textAlign ()`
- Alignement (vertical) : `textBaseline ()`
- Coupure de ligne : `wrapText ()`

- Informations : `measureText ()`

Canvas - Images

- Remplissage avec une image : `drawImage()`
- Paramètres variables :
  - Coordonnées
  - Coordonnées + Tailles (X et Y)
  - Coordonnées + Tailles (X et Y) dans image d'origine + Coordonnées + Tailles (X et Y) dans image finale
- Attention, l'image doit être chargée avant d'être utilisée (callback ou onload)

Canvas – Exercice 4

Canvas - Transformations

- Translation : `translate()`
- Rotation : `rotate()`
- Transformation : `transform()`
- Redimensionnement : `scale ()`
- Reset : `setTransform(1,0,0,1,0,0)`
- Mémorisation : `save ()` / `restore ()`

SVG

- Scalable Vector Graphics
- Format de graphiques vectoriel
- Basé sur XML

```
<svg width="1000" height="1000">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow"/>
</svg>
```

SVG - Formes

- Rectangle `<rect>`
- Circle `<circle>`
- Ellipse `<ellipse>`
- Line `<line>`
- Polyline `<polyline>`
- Polygon `<polygon>`
- Path `<path>`

SVG – Attributs Graphiques

fill	stroke	opacity
fill-opacity	stroke-opacity	
fill-rule	stroke-width	shape-rendering
	stroke-dasharray	image-rendering
marker	stroke-dashoffset	text-rendering
marker-end	stroke-linecap	color-rendering
marker-mid	stroke-linejoin	color-interpolation
marker-start	stroke-miterlimit	color-profile
	stop-opacity	stop-color

http://www.w3.org/TR/SVG/styling.html

SVG – Groupes

- Groupe <g>

```
<g fill="none" stroke="black" stroke-width="4" >
        <rect x="150" y="150" width="100" height="100" />
        <rect x="200" y="150" width="100" height="100" />
</g>
```

SVG - Rectangle

- Rectangle <rect>

```
<rect x="150" y="150" width="100" height="100" rx="20" ry="20"
fill="red" stroke="black" stroke-width="5" opacity="0.5" />
```

Attributs		
x	rx	width
y	ry	height
	style	

SVG - Cercle

- Circle `<circle>`

```
<circle cx="50" cy="350" r="50" stroke="black"  stroke-width="3" fill="red" />
```

Attributs		
cx	cy	r

SVG - Ellipse

- Ellipse `<ellipse>`

```
<ellipse cx="200" cy="350" rx="50" ry="20" fill="yellow" />
```

Attributs	
cx	rx
cy	ry

SVG - Line

- Line <line>

```
<line x1="300" y1="0" x2="400" y2="100" stroke="rgb(255,0,0)"
stroke-width="2" />
```

Attributs	
x1	x2
y1	y2

SVG - Polyline

- Polyline `<polyline>`

```
<polyline points="300,300 305,310 320,330 340,380 350,390 400,400"
fill="none" stroke="black" stroke-width="3" />
```

Attributs

points

SVG - Polygon

- Polygon `<polygon>`

```
<polygon points="450,300 455,310 470,330 490,380 500,390 550,400"
fill="none" stroke="black" stroke-width="3" />
```

Attributs

points

SVG - Path

- Path <path>

```
<path d="M 375 150 L 300 250 L 400 250 Z" />
```

Valeurs pour d		
M <X> <Y>	L <X> <Y>	Q
C <X> <Y>	H <X> <Y>	T
S <X> <Y>	V <X> <Y>	A
	Z	

SVG – Filter Effects

feBlend
feOffset
feFlood
feGaussianBlur
feImage
feMerge
feComposite
feColorMatrix
feConvolveMatrix
feTurbulence
feMorphology
feTile
feDisplacementMap
feDiffuseLighting
feSpecularLighting
feDistantLight
fePointLight
feSpotLight
feComponentTransfer

http://www.w3schools.com/svg/svg_filters_intro.asp

SVG – Filter Effects

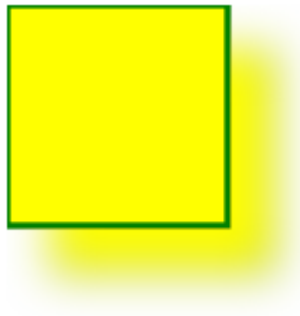
```
<defs>
     <filter id="f1" x="0" y="0" >
         <feGaussianBlur in="SourceGraphic" stdDeviation="15" />
     </filter>
</defs>
<rect x="600" y="0" width="100" height="100" stroke="green" stroke-width="3"
fill="yellow" filter="url(#f1)" />
```

SVG – Filter Effects

```
<defs>
    <filter id="f2" x="0" y="0" width="200%" height="200%" >
        <feOffset result="offOut" in="SourceGraphic" dx="20" dy="20" />
        <feBlend in="SourceGraphic" in2="offOut" mode="normal" />
    </filter>
</defs>
<rect x="750" y="0" width="100" height="100" stroke="green" stroke-width="3"
fill="yellow" filter="url(#f2)" />
```

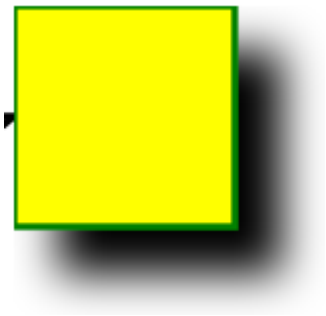
SVG – Filter Effects

```
<defs>
    <filter id="f3" x="0" y="0" width="200%" height="200%" >
        <feOffset result="offOut" in="SourceGraphic" dx="20" dy="20" />
        <feGaussianBlur result="blurOut" in="offOut" stdDeviation="10" />
        <feBlend in="SourceGraphic" in2="offOut" mode="normal" />
    </filter>
</defs>
<rect x="750" y="150" width="100" height="100" stroke="green" stroke-width="3"
fill="yellow" filter="url(#f3)" />
```



SVG – Filter Effects

```
<defs>
    <filter id="f4" x="0" y="0" width="200%" height="200%" >
        <feOffset result="offOut" in="SourceAlpha" dx="20" dy="20" />
        <feGaussianBlur result="blurOut" in="offOut" stdDeviation="10" />
        <feBlend in="SourceGraphic" in2="offOut" mode="normal" />
    </filter>
</defs>
<rect x="750" y="300" width="100" height="100" stroke="green" stroke-width="3"
fill="yellow" filter="url(#f4)" />
```



WebGL

- http://codepen.io/dissimulate/full/KrAwx
- http://madebyevan.com/webgl-water/

Crédits

Auteur

- Christophe Delagarde

Sources

- Mickaël Martin Nevot (www.mickael-martin-nevot.com)
- w3schools (http://www.w3schools.com/)
- HTML5 Canvas Tutorial (http://www.html5canvastutorials.com/)