

Examen de Contrôle Continu – Jeudi 29 novembre 2018

Durée : 3h

Avertissement : Les documents sont autorisés ainsi que l'accès à internet. Par contre l'utilisation d'outils de messagerie et/ou de discussion en ligne sont **strictement interdits**. De même tout échange de fichiers avec une tierce personne ou tout accès au répertoire d'un autre étudiant sera considéré comme **une fraude** et sanctionné en conséquence.

Les deux parties sont indépendantes. Vous pouvez donc les effectuer dans l'ordre qui vous convient.

Consignes

Cet examen se déroulant sur machine, vous devrez à la fin de l'épreuve remettre votre travail sous la forme d'un fichier compressé (zip ou tar/gz) que vous déposerez dans l'espace prévu à cet effet sur le serveur moodle de l'ufr im2ag. Afin qu'il n'y ait pas de confusion et de conflits lors de la remise de votre travail vous devrez **OBLIGATOIREMENT** respecter les consignes qui suivent.

1. Téléchargez sur votre compte le fichier [CC_PL2AI_Nov2018_XXX.zip](#) et décompressez le.

`unzip CC_PL2AI_Nov2018_XXX.zip`

Le répertoire `CC_PL2AI_Nov2018_XXX` ainsi créé à la structure suivante (fig. 1)

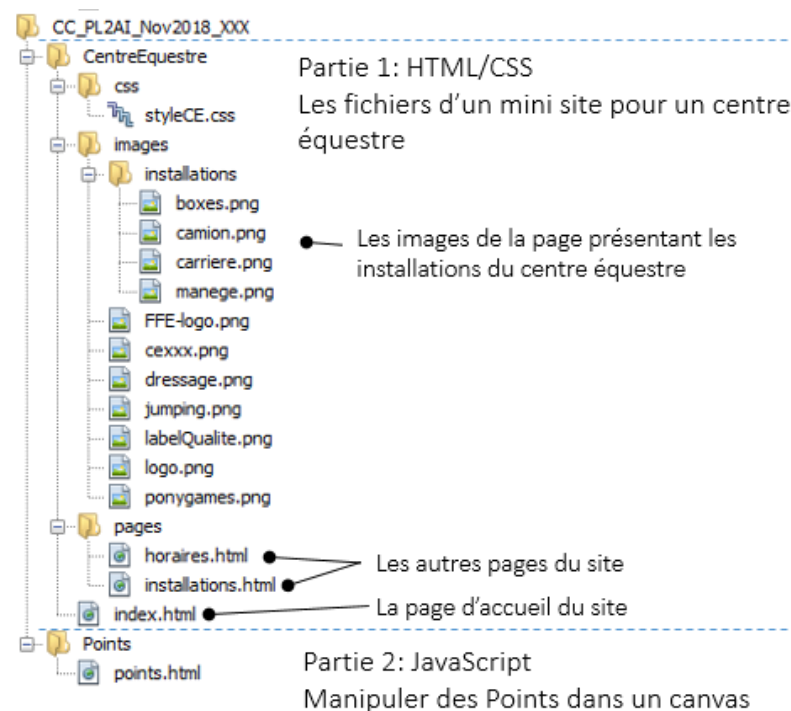


Fig.1: contenu du répertoire `CC_PL2AI_Nov2018_XXX`.

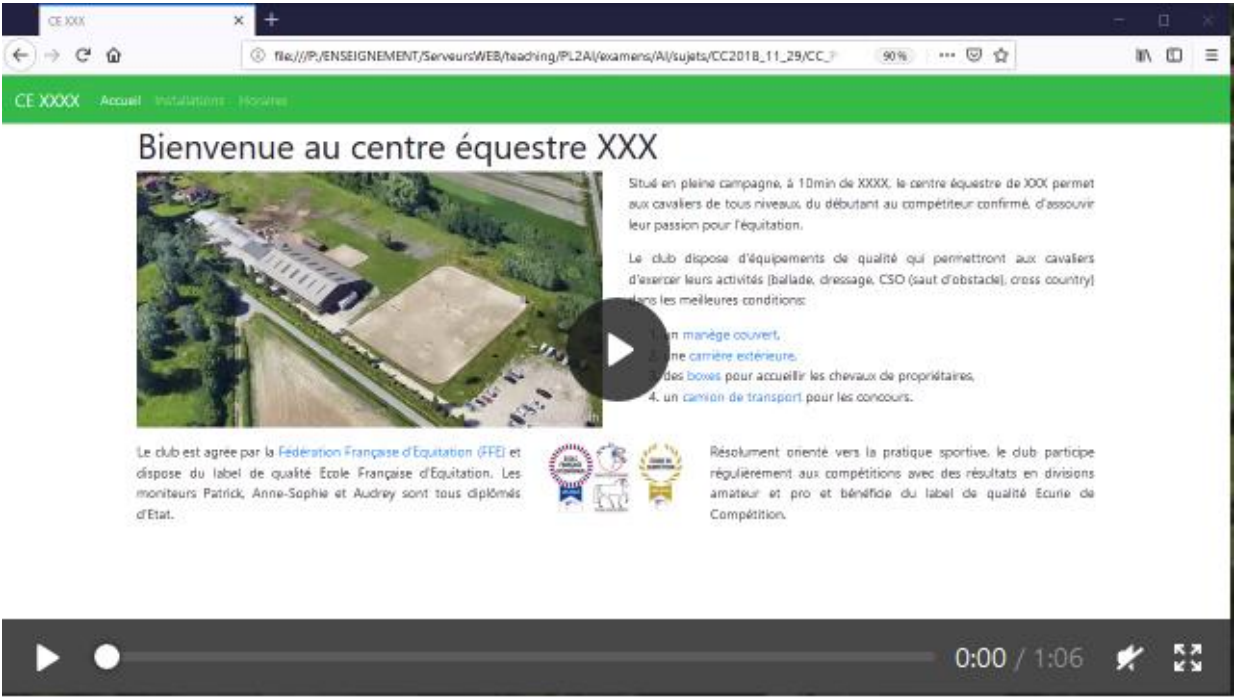
2. **Renommez** le répertoire `CC_PL2AI_Nov2018_XXX` en `CC_PL2AI_Nov2018_votreNom`, où `votreNom` correspond à votre identifiant de login (sans espaces, ni accents).

```
mv CC_PL2AI_Nov2018_XXX CC_PL2AI_Nov2018_votreNom
```

3. C'est le contenu de ce répertoire que vous devrez rendre en fin d'épreuve. **Soyez très vigilants à ce que tout le travail que vous effectuez durant l'épreuve soit bien enregistré dans ce répertoire.**
4. A la fin de l'épreuve **créez une archive compressée** de votre travail.
 1. placez-vous dans le répertoire contenant votre répertoire de travail `CC_PL2AI_Nov2018_votreNom`. Par exemple, si celui-ci est dans votre *home dir* : `cd` ou bien `cd ~`
 2. créez un fichier archive :
`tar zcvf CC_PL2AI_Nov2018_votreNom.tar.gz CC_PL2AI_Nov2018_votreNom` sous linux, ou bien fichier `.zip` sous windows.
5. **Déposez l'archive** sur le serveur moodle.
 1. Connectez-vous sur moodle
 2. si ce n'est déjà fait, enregistrez-vous au cours Programmation Internet,
 3. déposez le fichier `CC_PL2AI_Nov2018_votreNom.tar.gz` précédemment créé dans l'espace prévu à cet effet dans le moodle du cours Applications Internet.

Partie 1 : HTML / CSS / Bootstrap 4

Dans cet exercice il s'agit de reprendre le code HTML du site CentreEquestre afin de le compléter et de le finaliser et d'obtenir un affichage identique à celui proposé dans la vidéo ci-dessous.



Dans les deux premiers exercices vous ne vous occuperez pas de la mise en forme qui sera l'objet des exercices 3 et 4.

Exercice 1 (Tableaux HTML)

La page `horaires.html` donne les horaires d'ouverture du bureau du centre équestre et les heures des différentes leçons proposées. Pour le moment les horaires sont affichés sous forme de listes à puce. Pour une plus grande lisibilité, on souhaite les afficher sous forme d'un tableau comme indiqué sur la figure ci-dessous.

CE XXXX

- Accueil
- Installations
- Horaires

Horaires

Horaires du bureau

- tel: 04 76 XX XX XX
- courriel: cexxxx@mail.com
- heures d'ouverture du bureau
 - Mardi au Vendredi: 10h-12h, 15h-18h
 - Samedi: 8h30-12h, 14h-18h

horaires des cours

CSO

Débutant

- lundi: 17h30-18hH30
- mercredi: 15h-16h
- samedi: 14h-15h

Intermédiaire

- lundi: 17h30-18hH30
- mercredi: 17h-18h
- samedi: 15h-16h

Confirmé

- lundi: 18h30-19hH30
- mercredi: 18h-19h
- samedi: 17h-18h

Dressage

Débutant

- mardi: 16h-17h
- jeudi: 17h30-18h30

Intermédiaire

- mardi: 17h30-18hH30
- jeudi: 16h-18h
- samedi: 10h-11h

Confirmé

- mardi: 18h30-19hH30
- jeudi: 18h30-19hH30
- samedi: 11h30-12h

Pony-games

Confirmé

- samedi: 15h-17h

Activités	Niveau		
	Débutant	Intermédiaire	Confirmé
CSO	<ul style="list-style-type: none">• lundi: 17h30-18hH30• mercredi: 15h-16h• samedi: 14h-15h	<ul style="list-style-type: none">• lundi: 17h30-18hH30• mercredi: 17h-18h• samedi: 15h-16h	<ul style="list-style-type: none">• lundi: 18h30-19hH30• mercredi: 18h-19h• samedi: 17h-18h
Dressage	<ul style="list-style-type: none">• mardi: 16h-17h• jeudi: 17h30-18h30	<ul style="list-style-type: none">• mardi: 17h30-18hH30• jeudi: 16h-18h• samedi: 10h-11h	<ul style="list-style-type: none">• mardi: 18h30-19hH30• jeudi: 18h30-19hH30• samedi: 11h30-12h
Pony-games			<ul style="list-style-type: none">• samedi: 15h-17h

Mise en forme dans une table HTML

Pour afficher la bordure utiliser l'attribut `border="1px"` avec la l'élément `<table>`

figure 1: affichage des horaires dans une table HTML

Question 1 : en utilisant un tableau HTML, modifiez la page [horaires.html](#) de manière à obtenir l'affichage de la figure 1.

Dans ce premier exercice pour matérialiser les bordures du tableau utilisez l'attribut `border="1px"` pour l'élément `<table>`, vous occuperez de la mise en forme plus tard.

Exercice 2 (Liens hypertexte)

Toujours sans vous intéresser à la mise en forme des pages, il s'agit d'assurer la navigation entre les pages du site, comme indiqué sur la figure 2 ci-dessous.



figure 2 - les liens de navigation.

Question 2 : modifiez les pages [index.html](#) et [installations.html](#) en rajoutant les liens hypertextes qui permettent :

1. de passer aux pages présentant les installations et les horaires lorsque l'on clique sur les éléments du menu de navigation.
2. de retourner sur la page d'accueil ou d'aller à la page horaires depuis la page [installations.html](#).
3. d'aller à la présentation d'une installation spécifique lorsque l'on clique sur son nom.
4. d'aller vers la page du site de la FFE (<http://www.ffe.com/>) lorsque l'on clique sur le texte *Fédération Française d'Equitation*

Exercice 3 (CSS/Bootstrap 4)

Il s'agit, maintenant que le site est fonctionnel, d'assurer une mise en forme plus attrayante en utilisant les styles CSS de Bootstrap 4.

Question 3 (grille Bootstrap) : En utilisant les styles CSS proposés par Bootstrap 4, modifiez la page d'accueil du site ([index.html](#)) afin d'obtenir le comportement d'affichage décrit sur la figure 3 ci-dessous.

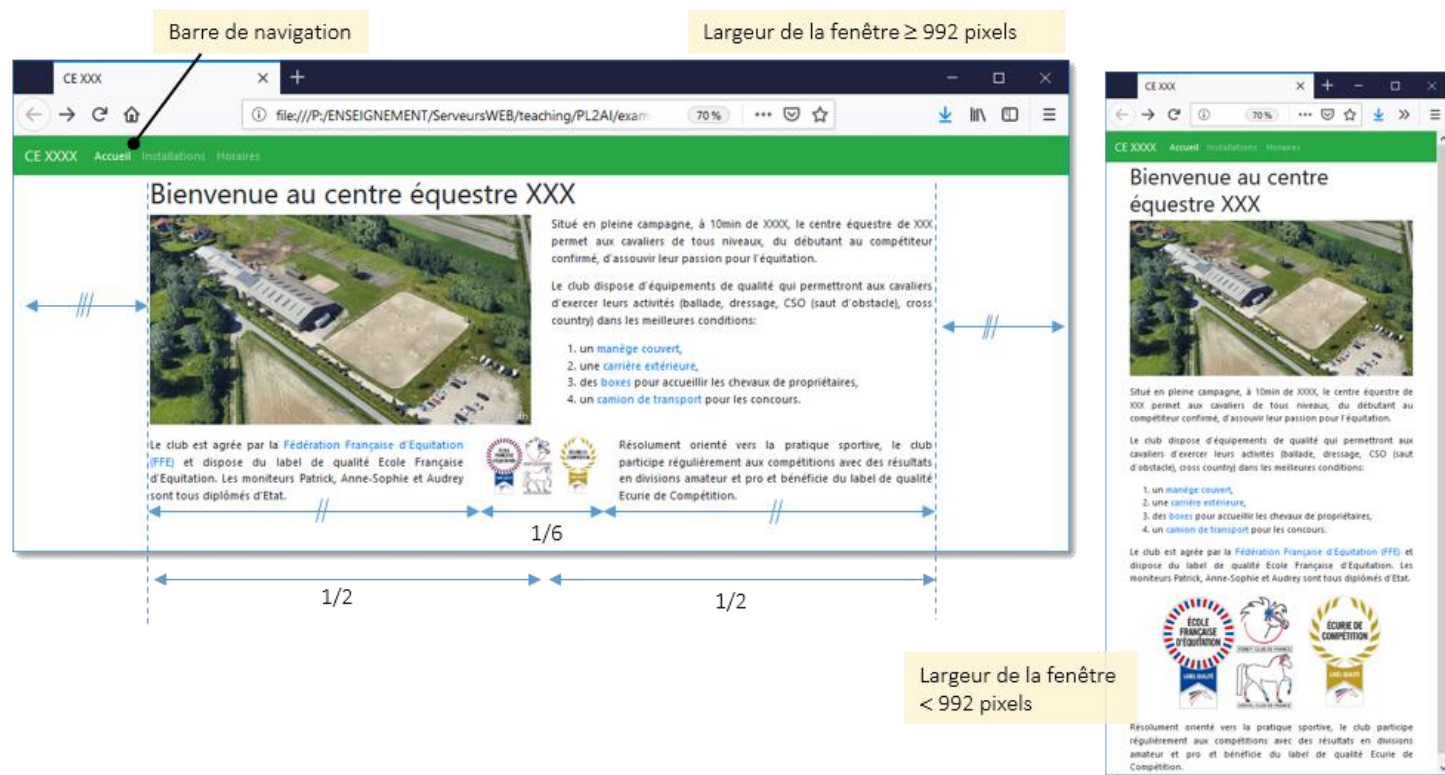


figure 3: affichage de la page d'accueil en utilisant Bootstrap 4.

1. Si la largeur de la fenêtre de navigation est supérieure ou égale à 992 pixels
 - le contenu de la page, hors la barre de navigation, soit centré sur la fenêtre du navigateur avec une marge gauche et une marge droite identiques,
 - l'image aérienne du centre équestre et la première section de texte s'affichent côte à côte, chacune occupant la moitié de la largeur utile, l'image adaptant automatiquement sa taille à la largeur de la fenêtre,
 - les deux dernières sections de texte soient de part et d'autre de l'image des labels *Ecole d'équitation* et *Ecurie de compétition*, celle-ci occupant 1/6 de la largeur utile.
2. si la largeur de la fenêtre de navigation est inférieure à 992 pixels les éléments se positionnent les uns au-dessous des autres.

Question 4 : (images et tableaux Bootstrap)



Question bonus, à ne faire que si il vous reste du temps.

Une fois la page d'accueil correctement stylée avec Bootstrap 4, faites de même avec les pages [installations.html](#) et [horaires.html](#) afin d'obtenir un site complet conforme à la [vidéo](#) présentée en début de cette partie.

1. Ajoutez aux deux pages la barre de navigation

2. Dans la page [installations.html](#) faites en sorte que les images s'affichent sous forme de vignette à gauche du texte comme sur la figure ci-dessous.

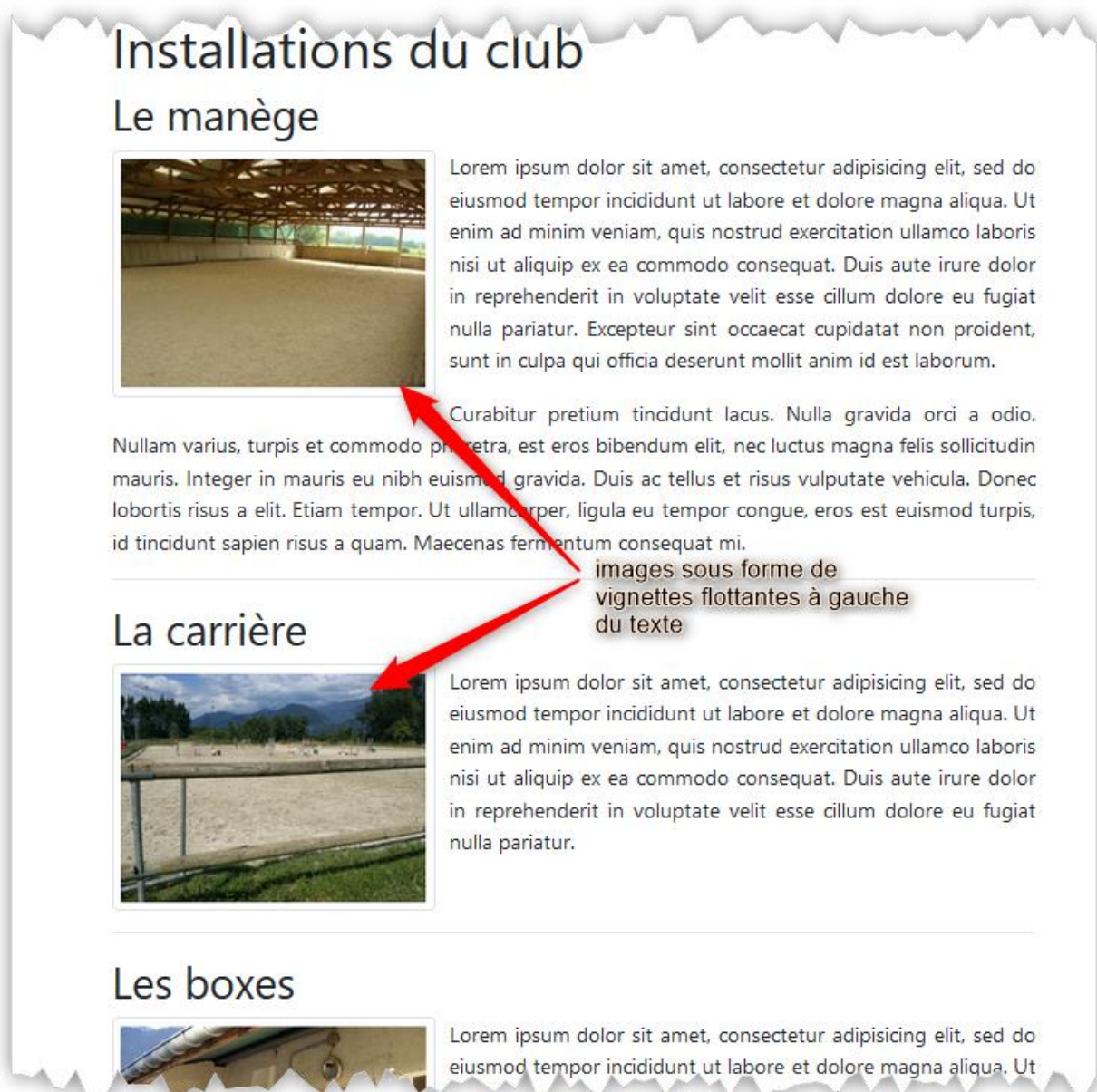


figure 4: affichage des images dans la page [installations.html](#).

- le [tutorial](#) de w3schools sur les images dans Bootstrap 4.
- la [documentation](#) de Bootstrap 4.
- Utilisez la propriété CSS `clear` pour indiquer si un élément peut être ou non situé à côté d'éléments flottants qui le précèdent.
 - le [tutorial](#) de w3schools.
 - la [documentation](#) de MDN

- Dans la page `horaires.html` faites en sorte que la table des horaires s'affiche comme sur la figure ci-dessous.

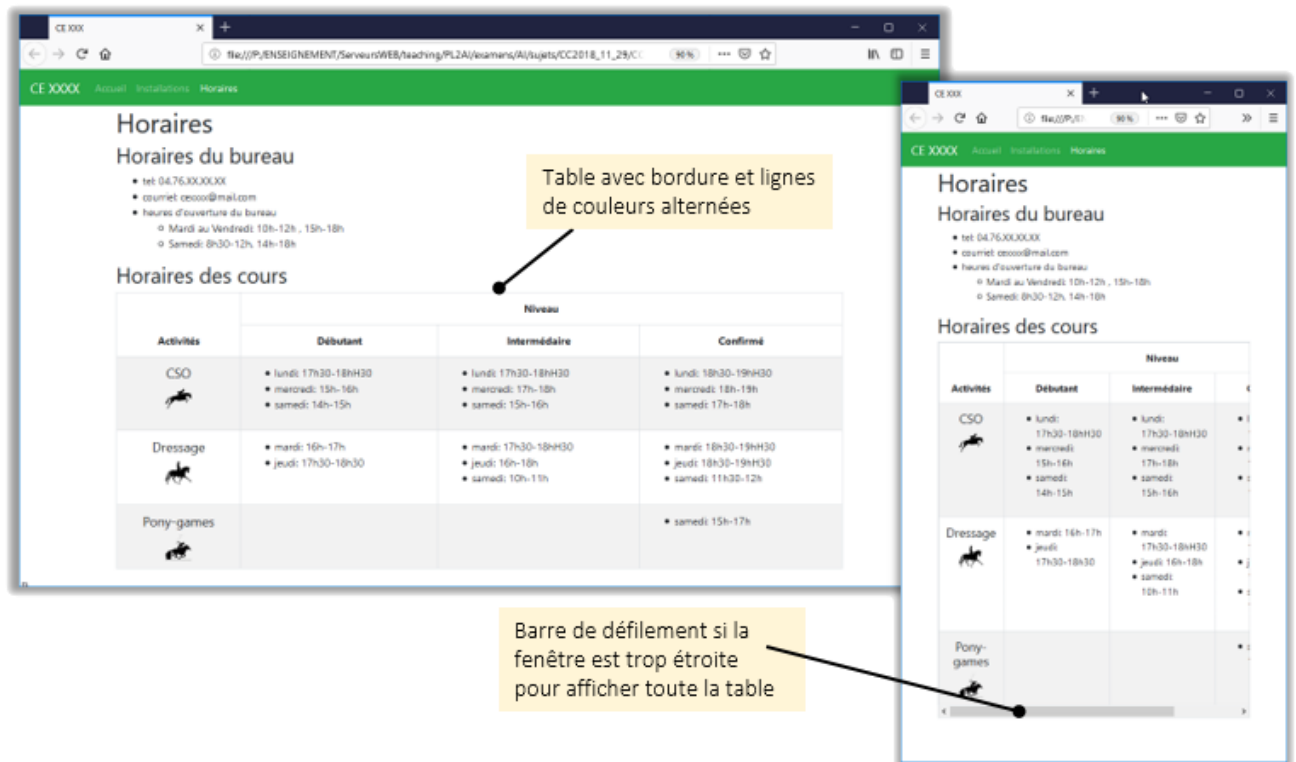


figure 5: affichage de la table des horaires dans la page `horaires.html`.

- le [tutorial](#) de w3schools sur les Tables.
- la [documentation](#) de Bootstrap 4.

Partie 2 : JavaScript

Dans cette partie il s'agit de programmer en JavaScript un nouveau type d'objets, `Point` qui représente un point du plan avec les attributs suivants :

- `nom` une chaîne de caractères correspondant au nom du point.
- `x, y` les coordonnées du point.
- `couleur` une chaîne correspondant à la couleur du point, la valeur par défaut étant le vert.

Exercice 1: programmer un type objet simple

Question 1 : dans un fichier `point.js` écrivez le code JavaScript pour le type `Point` à savoir:

- une fonction constructeur `Point(nom,x,y)` qui permet de créer un point nommé.
- une méthode `dessiner(ctxt)` qui permet de dessiner le point dans un canvas (`ctxt` étant le contexte graphique associé au canvas). Le dessin d'un point s'effectue en le matérialisant pas un disque centré sur les coordonnées du point, de rayon 4 pixels et rempli avec la couleur du point.

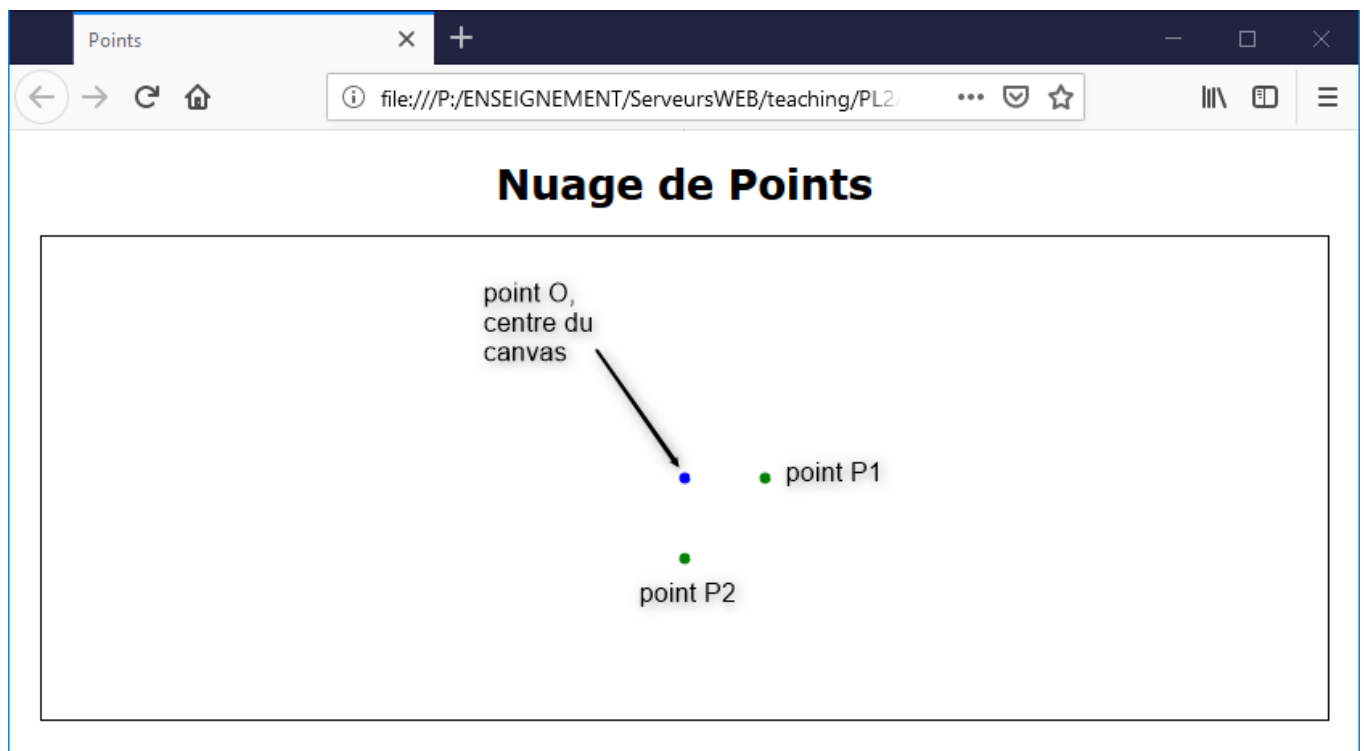
Pour dessiner le disque vous pouvez utiliser la méthode `arc` de l'objet `CanvasRenderingContext2D` associé au canvas.

- le [tutorial](#) de w3schools.
- la [documentation](#) de MDN.

Pour tester votre type `Point`, écrivez dans un fichier `lesPoints.js` un code JavaScript qui, une fois la page HTML `points.html` chargée, permet de :

- créer un point de nom `O`, dont les coordonnées correspondent au centre du canvas défini dans la page HTML, et de lui associer la couleur bleue,
- créer deux points de nom `P1` et `P2` respectivement à droite et au-dessous du point `O` à une distance de 40 pixels,
- d'afficher ces points dans le canvas.

Vous devez obtenir l'affichage suivant (Attention : vous ne devez afficher que les points et pas leur nom, le texte sur la figure n'est là que pour vous indiquer la position des points) :



Exercice 2 : utiliser les tableaux

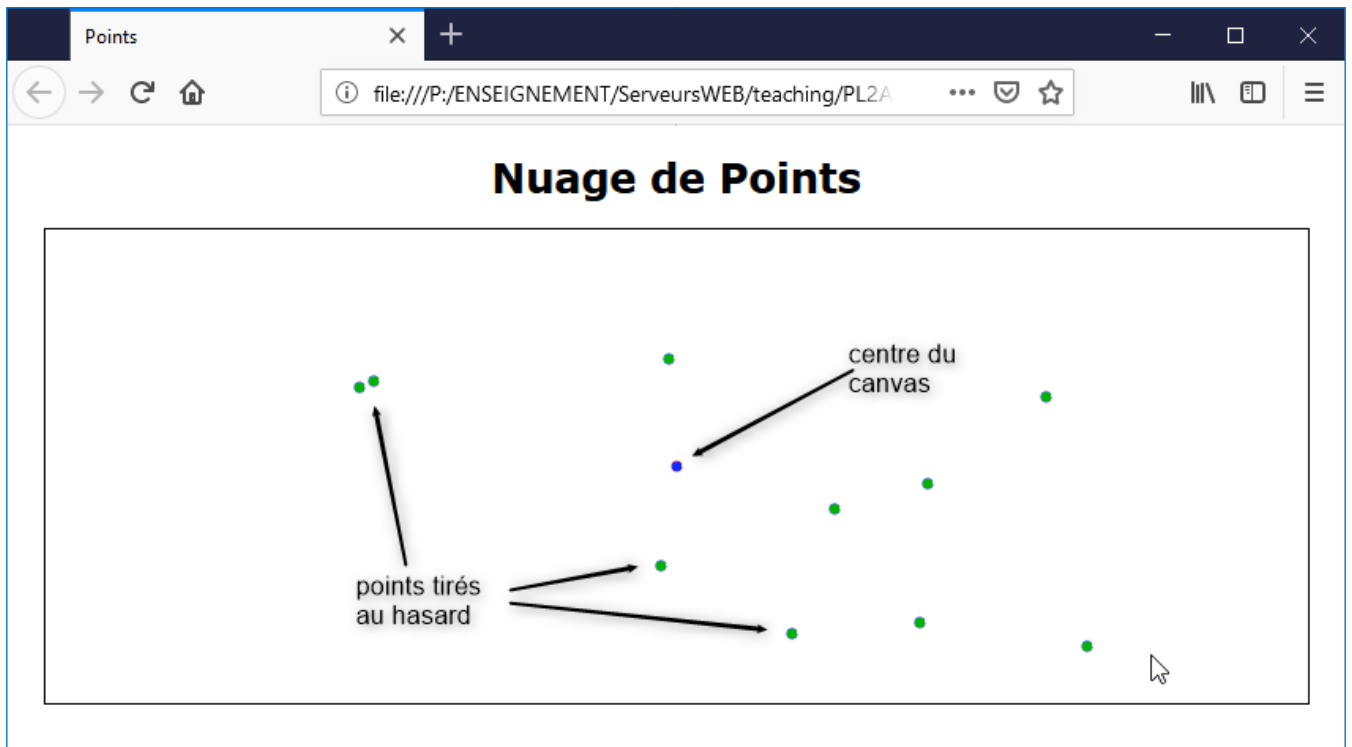
Question 2 : modifiez le fichier `lesPoints.js` afin que, une fois la page HTML `points.html` chargée il permette de :

- créer un point `O`, dont les coordonnées correspondent au centre du canvas défini dans la page HTML.
- créer une liste de 10 points, nommés `P1`, `P2`, ..., `P10` dont les coordonnées sont tirées au hasard à l'intérieur du canvas défini dans la page HTML. Ces points seront stockés dans un tableau.

Pour obtenir en JavaScript un nombre au hasard vous pouvez utiliser la fonction JavaScript `random` de l'objet `Math`.

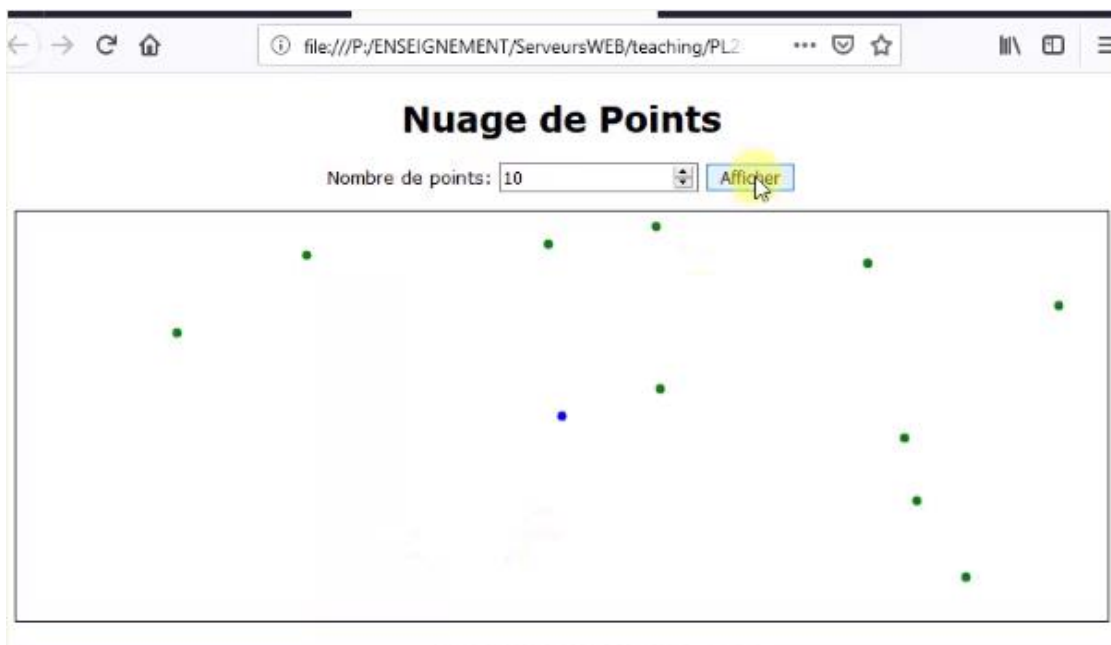
- le [tutorial](#) de w3schools.
- la [documentation](#) de MDN.

- afficher le point **O** en bleu et les points **P1, P2, ..., P10** (en vert) comme le montre la figure ci-dessous.



Exercice 3 : interagir avec la page HTML

Question 3 : modifiez la page `point.html` et le code du programme `lesPoints.js` afin de pouvoir fixer interactivement le nombre de points à afficher comme le montre la vidéo ci-dessous.



Exercice 4 : Etendre le type Point

Question 4 (ajouter des méthodes à Point) : rajoutez à votre type `Point`, deux nouvelles méthodes (au contraire de la méthode `dessiner`, leurs éventuels paramètres ne vous sont pas indiqués, cela sera à vous de les définir) :

- `toString` qui renvoie une chaîne de caractères représentant le point. Pour un point de nom `A` et de coordonnées `x=100` et `y=54` la chaîne retournée sera `A(100,54)`.
- `distanceA` qui permet de calculer et retourner la distance entre le point et un autre point.

Pour calculer en JavaScript la racine carré d'un nombre: la fonction javascript `Math.sqrt()`

La distance entre un point $P1(x1,y1)$ et un point $P2(x2,y2)$ est $\sqrt{(x2 - x1)^2 \times (y2 - y1)^2}$

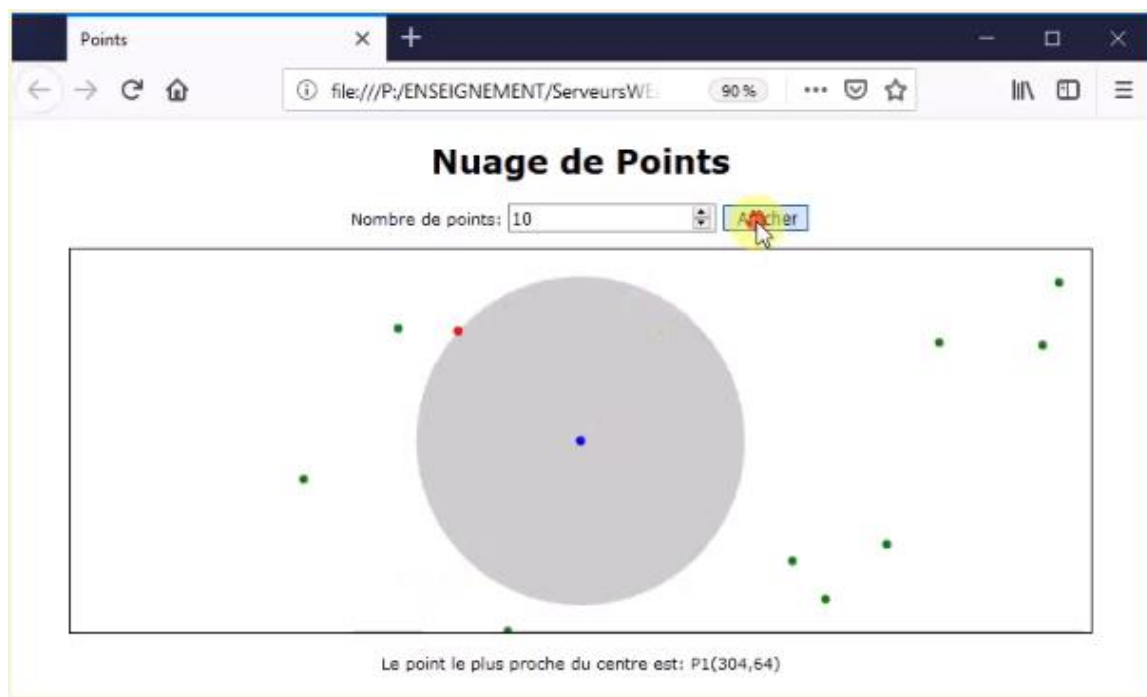
Pour calculer en JavaScript la racine carré d'un nombre vous pouvez utiliser la fonction JavaScript `sqrt` de l'objet `Math`.


- le [tutorial](#) de w3schools.
- la [documentation](#) de MDN.

Question 5 (chercher le point le plus proche du centre) : Pour tester vos méthodes `toString` et `distance` modifiez votre programme de `lesPoints.js` pour :

- trouver le point le plus proche du centre du canvas et l'afficher en rouge,
- afficher un disque gris passant par ce point et le centre du canvas,
- afficher sous le canvas le nom du point et ses coordonnées.

La vidéo ci-dessous vous montre le fonctionnement attendu du programme.



 AI - M2PCCI by Philippe Genoud is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).