

Contenu : - Modalités de l'interaction entre l'homme et la machine
Capacités attendues : - Identifier les différents composants graphiques permettant d'interagir avec une application Web.

1. Les événements liés à la souris

ANNEXE 1

La programmation d'aujourd'hui est une programmation **I**nterface **H**omme **M**achine qui utilise des événements pour piloter le comportement des programmes. Ce point est fondamental, la conception des IHM est construite autour de la réponse à certains événements, un appui sur une touche ou bien une combinaison de touches du clavier, un événement lié à la souris etc...

Une IHM est la partie d'un programme qui gère les échanges d'informations avec l'utilisateur. Cette partie est très importante mais elle est trop souvent négligée...

Quelques événements liés à la souris sous Windows

| Événement de souris | Description |
|---------------------|--|
| Click | Cet événement se produit lorsque vous relâchez le bouton de souris (en général avant l'événement <code>MouseUp</code>). Le gestionnaire de cet événement reçoit un argument de type <code>EventArgs</code> . Utilisez cet événement lorsque vous devez seulement déterminer à quel moment un clic se produit. |
| MouseDown | Cet événement se produit lorsque l'utilisateur clique sur le contrôle avec la souris. Le gestionnaire de cet événement reçoit un argument de type <code>MouseEventArgs</code> . Utilisez cet événement lorsque vous devez obtenir des informations à propos de la souris au moment d'un clic. |
| DoubleClick | Cet événement se produit lorsque vous double-cliquez sur le contrôle. Le gestionnaire de cet événement reçoit un argument de type <code>EventArgs</code> . Utilisez cet événement lorsque vous devez seulement déterminer à quel moment un double-clic se produit. |
| MouseDoubleClick | Cet événement se produit lorsque l'utilisateur double-clique sur le contrôle avec la souris. Le gestionnaire de cet événement reçoit un argument de type <code>MouseEventArgs</code> . Utilisez cet événement lorsque vous devez obtenir des informations à propos de la souris au moment d'un double-clic. |
| MouseEnter | Cet événement se produit lorsque le pointeur de la souris se trouve sur le contrôle et que l'utilisateur appuie sur un bouton de souris. Le gestionnaire de cet événement reçoit un argument de type <code>MouseEventArgs</code> . |
| MouseEnter | Cet événement se produit lorsque le pointeur de la souris entre dans la bordure ou dans la zone cliente du contrôle, selon le type de contrôle concerné. Le gestionnaire de cet événement reçoit un argument de type <code>EventArgs</code> . |
| MouseHover | Cet événement se produit lorsque le pointeur de la souris s'arrête et reste sur le contrôle. Le gestionnaire de cet événement reçoit un argument de type <code>EventArgs</code> . |

2. Comment gérer la taille des caractères

ANNEXE 2

Quelques éléments sur la gestion des tailles de caractères.

Nos pages web sont affichées sur des écrans d'appareils très différents les uns des autres depuis le PC de bureau jusqu'au mobile. Il faut en tenir compte. Sur le Web, les tailles de caractères ne sont pas gravées dans le marbre, et elles peuvent changer d'un ordinateur à l'autre, parfois du tout au tout.

En fonction de la taille de l'écran de l'internaute, de son système d'exploitation et de son navigateur, les caractères peuvent devenir tellement petits qu'ils en sont illisibles, ou aussi grands que dans des méthodes de lecture pour enfants. Les internautes ont heureusement la possibilité de choisir une taille "confortable" dans les préférences de leurs navigateurs, et donc de réduire un peu ces variations de taille, mais le résultat peut rester très différent de ce que vous attendiez.

Le design des pages doit s'adapter aux nouvelles technologies, elles doivent s'adapter aux différentes tailles d'écrans on parle de maquettes *liquides*.

Taille de caractères relatives

Si on souhaite que les pages touchent le plus large public, mieux vaut assigner une taille de caractères relative à la taille que le visiteur aura choisie par défaut. Il y a plusieurs façons d'assigner des tailles de caractères relatives.

Vous pouvez utiliser des pourcentages de la valeur par défaut (%), ou bien une unité nommée "em", qui correspond à 100%. 1.2em revient à 120%.

La meilleure méthode est d'utiliser un ensemble de descriptions pré-définies qui font penser à des tailles de tee-shirts. "Medium" revient à 100%, ou 1em. "Smaller" est une taille en-dessous, et "Larger" une taille au-dessus.

Il y a aussi x-small, xx-small, x-large et xx-large.

L'intérêt d'utiliser ce système par rapport aux pourcentages ou aux ems, c'est que les navigateurs ne laisseront pas la police d'écriture devenir si petite qu'on ne puisse plus la lire. Imaginons, par exemple, que l'on indique une taille de caractères de 0.7em ou de 70%

Superta phrase !
Super ta phrase !

qui rendrait très bien sur un PC. Pour quelqu'un équipé d'un Mac avec une taille de caractères par défaut plus petite, ce 70% pourrait faire passer vos caractères sous le seuil de la lisibilité. xx-small est une bonne solution pour déterminer quelle est la taille de caractères minimum que l'ordinateur peut afficher.

xx-large x-large large medium small x-small

xx-small

Caractères xx-small : C'est petit mais le texte reste lisible

Avec une taille fixe trop petite : C'est beaucoup trop petit

Résumé des tailles de caractères

- Le *px* (pixel). L'unité de mesure de votre écran. Par exemple 1024x768 pixels, un pixel est égal à un point lumineux de votre écran.
- Le *pt* (point), correspond à la taille des caractères imprimés. La taille standard d'une police de type Times est de 12 points.
- Le *pourcent* (%). Vous demandez à tel élément d'avoir telle taille en pourcentage par rapport à un autre élément. Par exemple affiche cet objet sur 50% de la largeur de la fenêtre.
- L'*em*. Un « em » est égal à la hauteur de la police dans sa taille normale. « Deux em » signifie donc une taille deux fois plus haute qu'une police en taille normale, « 0.5em » vaudra une demi taille, etc.
Le séparateur de décimale est le point, pas la virgule ! Ne l'oubliez pas.
- *Small, x-small, etc.* plus spécifiquement destiné à indiquer la taille des polices. Les valeurs possibles sont : **xx-small, x-small, small, medium, large, x-large, xx-large** :

Tailles de caractères absolues

Si votre site ne concerne qu'un nombre limité de visiteurs qui utilisent à peu près le même type d'ordinateur que vous, vous pouvez également indiquer des tailles de caractères en pixels.

Les pixels ne varient pas tellement d'un ordinateur de bureau à l'autre, et la taille des caractères sera similaire, sinon identique. Dès que l'on passe à des ordinateurs portables ou bien de poche, ou encore à des moniteurs très grands et à haute résolution, les tailles de pixels s'éloignent de la norme, et la taille des caractères sera modifiée d'autant.

Gestion des interlignes

La quantité d'espace entre les lignes est d'environ 120% de la taille de caractères par défaut. On peut en ajouter un peu plus, ce qui améliore généralement la lisibilité, surtout si les lignes sont longues. J'évoquerai la longueur des lignes plus tard.

Pour changer l'interligne, nous avons à nouveau l'option relative (en % ou en em) ou absolue (en px).

Dans l'exemple de code ci-dessous il est indiqué une hauteur de ligne (line-height), c'est-à-dire la hauteur du caractère et de l'espace supplémentaire au-dessus, à 180% de la taille de caractères (small) pour toute la page.

```
body {
background-color: #e8eae8;
color: #5d665b;
margin: 50px;
font-family: Verdana, Geneva, Arial, sans-serif;
font-size: small;
line-height: 180%
}
```

3. Bon usage des balises html (règles définies en xHTML et reprises par)

ANNEXE 3

- **Règle 1** : Écrire les balises en **minuscule**.
- **Règle 2** : Toutes les balises **ouvertes** par `<nom_balise>` doivent être **fermées** par `</nom_balise>` sauf s'il s'agit d'une balise unique `<balise_unique />`.
- **Règle 3** : Les balises doivent être correctement **imbriquées** et **indentées**

Exemple : Dans le code ci-dessous, le `<a>` est **imbriquée** avec `<p></p>`, `<div>` et `<p>` sont **indentés**.

```
<div>
  <p><a href="page2.html">Oeuvre</a>
  </p>
</div>
```

- **Règle 4** : Les valeurs des **attributs** doivent toujours figurer entre des **guillemets**.

Exemple : `< a href="bibliographie.html">`

4. Le positionnement en CSS

ANNEXE 4

- En HTML, la plupart des balises peuvent se ranger dans l'une ou l'autre des **catégories** : **block** et **inline** :

- **block** : une balise de type "block" crée automatiquement un **retour à la ligne avant et après**.

¹ Site <http://www.pompage.net/traduction/cssdezero-3>

Une page web est constituée d'un ensemble de blocs que l'on peut mettre les uns à l'intérieur des autres.

Exemple de balise de type block : les titres `<h1></h1>`, les paragraphes `<p></p>`

block

`<h1> Titre (block) </h1>`

- **inline** : une balise de type "inline" se trouve **obligatoirement à l'intérieur** d'une balise de type block.

Une balise de type inline ne crée pas de retour à la ligne.

Exemple de balise de type inline.

inline

`<p> Paragraphe blablablaba
<a> Lien inline </p>`

C'est par exemple le cas des liens : `<a>`

- **Transformation d'un élément avec "display"**

La propriété CSS **display** permet de transformer n'importe quel élément d'un type dans un autre.

display

| Valeur | Exemples | Description |
|---------------------|--|--|
| inline | <code><a></code> , <code></code> , <code></code> ... | Éléments d'une ligne. Se placent les uns à côté des autres. |
| block | <code><p></code> , <code><div></code> , <code><section></code> ... | Éléments en forme de blocs. Se placent les uns en dessous des autres et peuvent être redimensionnés. |
| inline-block | <code><select></code> , <code><input></code> | Éléments positionnés les uns à côté des autres (comme les inlines) mais qui peuvent être redimensionnés (comme les blocs). |
| none | <code><head></code> | Éléments non affichés. |

Exemple : Positionnement des liens les uns en dessous des autres et redimensionnement possible.

```
a {
  display : block ;
}
```

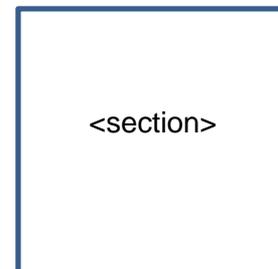
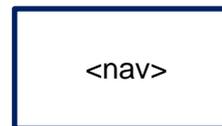
- **Positionnement inline-block**

Les éléments de type **inline-block** se positionnent **les uns à côté des autres** et on peut leur donner des dimensions précises.

Exemple : positionnement d'un **bloc nav** à côté d'un bloc section.

```
.nav {
  display : inline-block ;
  width : 150px ;
  border: 1px solid black;
  vertical-align: top;
}

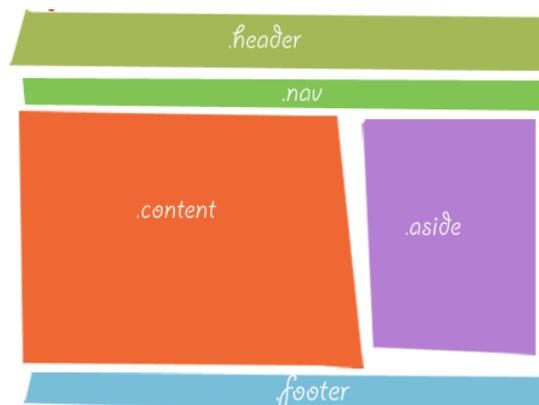
.section {
  display : inline-block ;
  border: 1px solid black;
}
```



5. Structure HTML de base (squelette)

ANNEXE 5

Le **squelette "de base"** d'une page web peut ressembler à ceci :



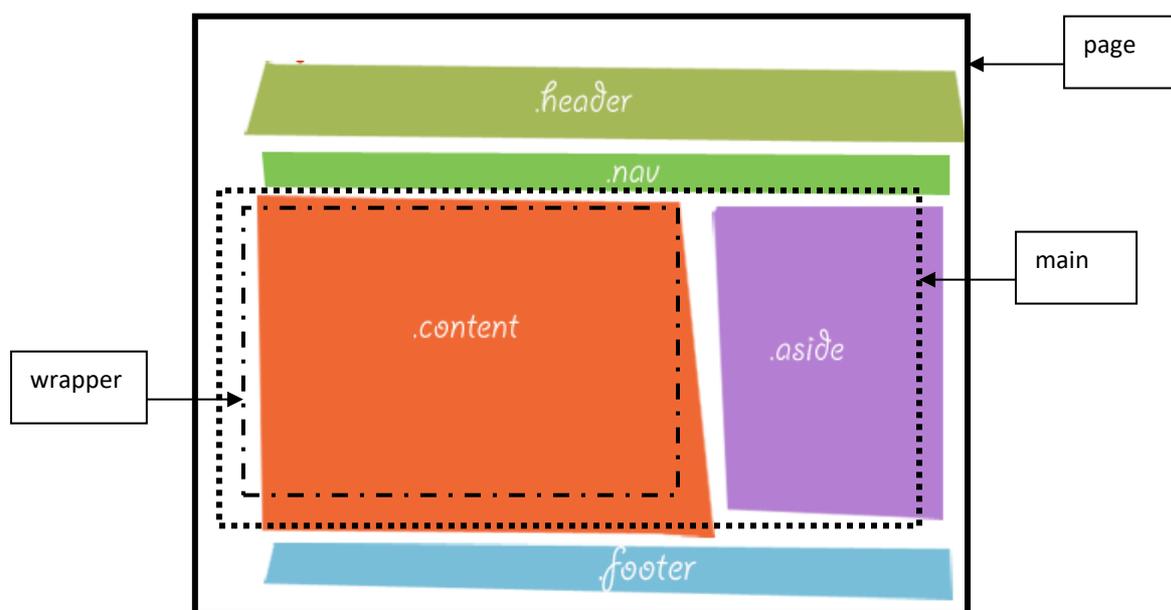
"Cette structure à **cinq blocs principaux** convient dans la majorité des cas, car elle **permet de fabriquer une grande variété** de mises en page. Bien que ce soit conceptuellement très simple, ça reste toujours assez **délicat à mettre en page en HTML et CSS.**"
 Codage du squelette de la page ci-dessus :

| HTML | CSS |
|---|--|
| <pre><!DOCTYPE html> <html> <head> <title>Titre de la page</title> <link rel="stylesheet" href="style.css"/> </head> <body> <div class="header">En-tête</div> <div class="nav">Navigation</div> <div class="content">Contenu</div> <div class="aside">Contexte</div> <div class="footer">Pied de page</div> </body> </html></pre> | <pre>/* Mes styles */ .header {...} .nav {...} .content {...} .aside {...} .footer {...}</pre> |

Remarque : pour être vraiment à l'aise en intégration CSS, il est utile d'ajouter quelques `<div>` englobantes :

- **page** (parfois appelé `.container`) **englobe toute la page**, comme son nom l'indique
- **main** contient les blocs centraux, ici `.content` et `.aside`, mais qui pourrait être plus nombreux.
- **wrapper** englobe le `.content` seul, juste pour **faciliter la mise en page.** »¹

Présentation de la page web en tenant compte de la remarque précédente.



Codage du **layout** de la page en tenant compte de la remarque ci-dessus :

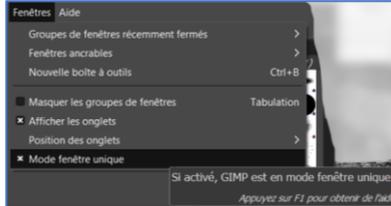
| HTML | CSS |
|---|---|
| <pre><!DOCTYPE html> <html> <head> <title>Titre de la page</title> <link rel="stylesheet" href="style.css"/> </head> <body> <div class="page"> <div class="header">En-tête</div> <div class="nav">Navigation</div> <div class="main"> <div class="wrapper"> <div class="content">Contenu</div> </div> <div class="aside">Contexte</div> </div> </div></pre> | <pre>/* Mes styles */ .page {...} .header {...} .nav {...} .main {...} .wrapper {...} .content {...} .aside {...} .footer {...}</pre> |

Nous allons redimensionner la photo "Calamar-grand-dessin.jpg" pour l'insérer dans notre page web. Les étapes à suivre sont listées ci-après :

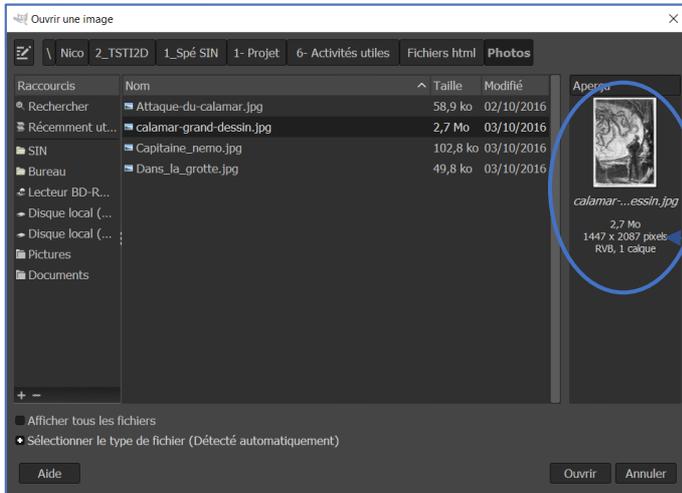
1- Ouvrir le logiciel Gimp



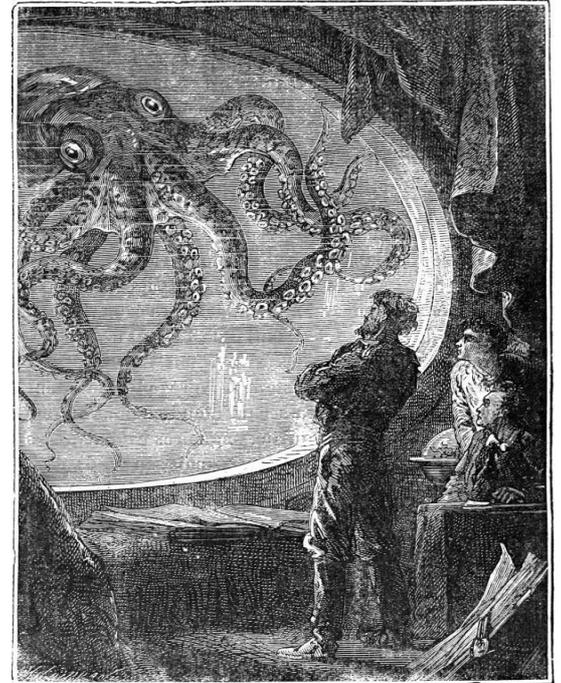
2- Se mettre en mode fenêtre unique



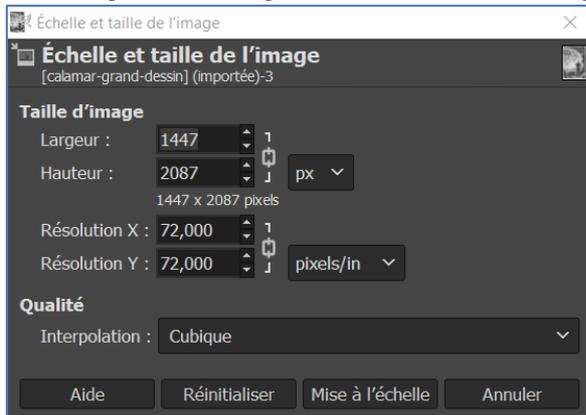
3- Charger la photo : fichier -> Ouvrir



Au passage nous obtenons la dimension de l'image



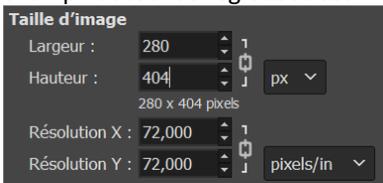
4- Pour redimensionner l'image : Menu Image -> Échelle et taille de l'image :



Les proportions seront conservées si le petit cadenas est fermé :



5- Dans cette fenêtre, il suffit de définir l'une des dimensions ici la largeur à 280 pixels. On tape 280 dans le champ largeur et le champ hauteur se règle tout seul :

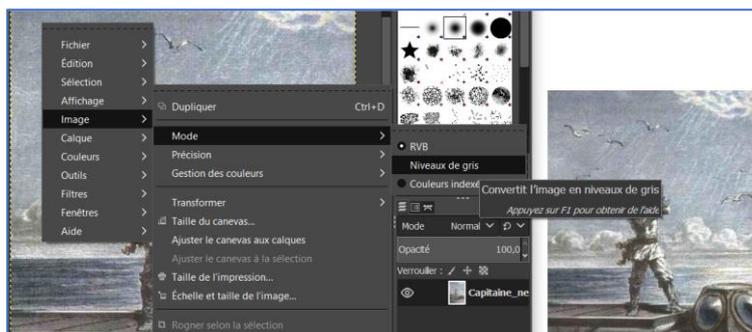


6- Pour terminer cliquer sur *Mise à l'échelle*.

7- Sauvegarder l'image par la commande *Exporter sous* et la nommer *calamar-petit-dessin.jpg*.

b. Mettre une image en noir et blanc

Nous souhaitons afficher nos illustrations en noir et blanc. L'image du capitaine Nemo est en couleur. Faire un clic droit sur l'image puis choisir niveau de gris :



Il ne reste plus qu'à enregistrer (*Fichier -> Exporter Sous* pour conserver le format .jpg) le résultat.

1. Eco-conception d'un site WEB²

ANNEXE 8

Pourquoi se préoccuper de l'éco-conception d'un site

Il fut un temps où la qualité d'un code se mesurait à ses performances par rapport aux ressources utilisées. C'était l'époque où les capacités des processeurs, la quantité de mémoire vive ou les capacités de stockage constituaient de véritables limites qui étaient prises en compte. Un temps où les ordinateurs prenaient de la place sur le bureau et y restaient longtemps (dans la mémoire du porte-monnaie aussi). Une époque où l'on reconnaissait un bon développeur à ses aptitudes à faire rentrer son code dans la petite zone allouée. Ces choses étaient précieuses. C'était il y a quelques dizaines d'années.



Et puis, alors que le changement climatique est devenu une réalité quotidienne dans la vie de millions d'humains, alors que les métaux rares et autres ressources non renouvelables sont à l'origine de conflits géopolitiques, alors que les enseignements scolaires de nos pays mettent en avant les questions du développement durable et prônent notre soi-disant "exemplarité" en matière de réduction de gaz à effet de serre, on assiste tout simplement à un engouement pour un développement logiciel où l'efficacité se mesure en temps de travail pour l'informaticien et non plus en efficacité par rapport aux ressources.

Et tant pis si cette précipitation se traduit par des logiciels toujours plus gourmands, puisque les ressources sont perçues comme "virtuelles" et donc "illimitées" : quand ça "rame", il suffit de remplacer son Smartphone ou sa tablette par un modèle plus puissant et le tour est joué. De toute façon, le code sera obsolète à la prochaine génération d'équipements et il faudra donc le réécrire rapidement.

Les conséquences ne sont pas visibles à l'échelle de temps du développement logiciel : pourquoi se préoccuper d'autre chose ?

Et pendant ce temps, le gaspillage fait des ravages. Dans ce monde où l'information nous submerge, où les sites web rivalisent de générosité dans une ambiance générale du "tout gratuit" et du toujours plus, il n'y a plus de temps et, "à force d'oublier l'essentiel pour l'urgence, de faire de l'urgence l'essentiel, on finit par oublier l'urgence de l'essentiel" (Edgar Morin).

Comment éco-concevoir son site web

Tout d'abord comprendre les mécanismes en jeu dans les échanges sur le web entre le poste client et le serveur. Tout a un coût : l'espace stocké chez un hébergeur, le flux de données effectuant des allers retours pour répondre à une requête, la consommation de bande passante sur les lignes, le temps de cycle processeur nécessaire au serveur pour calculer la réponse ...

Alors il faut se documenter comme par exemple en visitant ce site : <http://backbn.fr/bonnes-pratiques/> par exemple.

² Frédéric BORDAGE, Eco-conception web / Les 115 bonnes pratiques, 2^{ème} édition, Eyrolles, 2015, 168 pages.