

<p>T_{erm} </p>	<p align="center">Initiation à la Programmation Orientée Objet (POO) illustrée avec jQuery</p>	
<p align="center">Cours / TD prof</p>	<p>Mots clé : objet, état, attribut, valeur, méthode, constructeur, évènement.</p>	

Centre d'intérêt

CI4 : Gestion de l'information / Structures matérielles et Logicielles

TP associés : TP5 « Page Web » : "Afficher des données dans une page web"



Table des matières

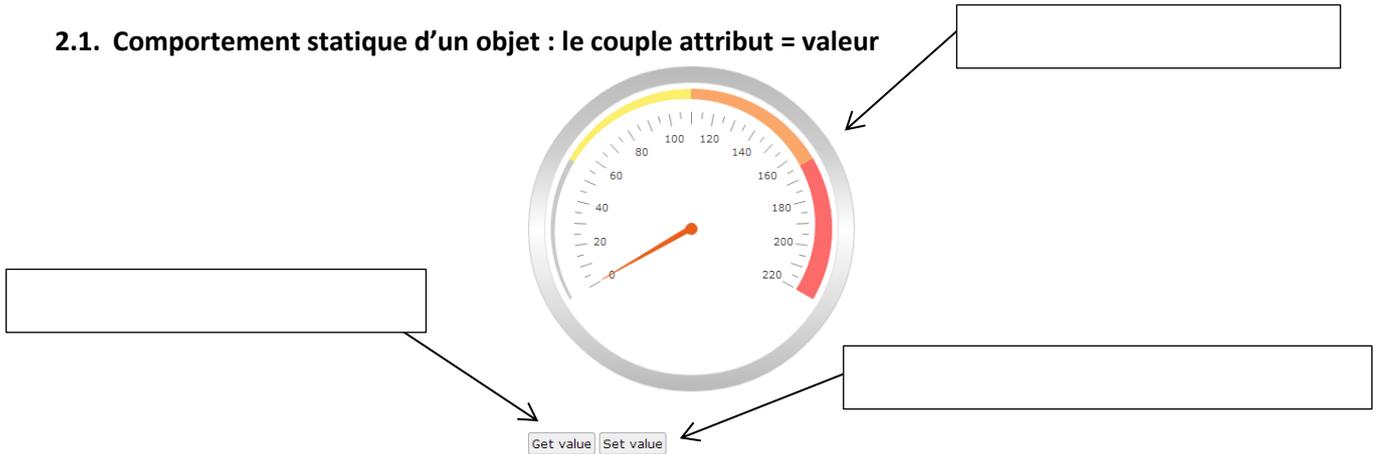
1. Une définition	2
2. Approche pratique de la notion d'objet et de programmation orientée objet	2
2.1. Comportement statique d'un objet : le couple attribut = valeur	2
2.2. Construction de l'objet et affichage	3
2.3. Comportement dynamique d'un objet : application d'une méthode	4
2.4. Les évènements.....	4
3. JQuery est une bibliothèque Javascript	4
A retenir	4
Annexe 1 : Documentation de l'objet "Gauge" (JQWidget).....	5
Annexe 2 : Mise en œuvre de l'objet "Gauge" (JQWidget) avec un script JQuery	6

1. Une définition

« La **programmation orientée objet (POO)**, ou **programmation par objet**, est un paradigme¹ de programmation informatique élaboré par les norvégiens Ole-Johan Dahl et Kristen Nygaard au début des années 60 et poursuivi par les travaux d'Alan Kay dans les années 1970. Il consiste en la définition et l'interaction de **briques logicielles** appelées **objets** ; un objet représente un concept², une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une **structure interne** et un **comportement**, et il sait **communiquer** avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; la communication entre les objets via leurs relations permet de réaliser les fonctionnalités attendues, de résoudre le ou les problèmes. » **Wikipédia**

2. Approche pratique de la notion d'objet et de programmation orientée objet

2.1. Comportement statique d'un objet : le couple attribut = valeur



Extrait de code : jauge et boutons ci-dessus.

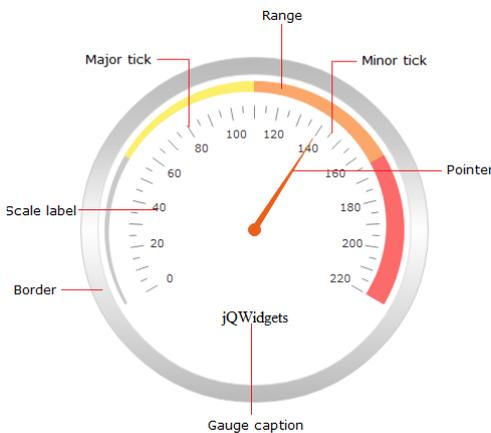
```

$( '#gaugeContainer' ).jqxGauge({
    ticksMinor: { interval: 5, size: '5%' },
    ticksMajor: { interval: 10, size: '9%' },
    value: 0,
    colorScheme: 'scheme03',
    animationDuration: 1200
});

$( '#setValueButton' ).jqxButton({ width: 70, theme: 'classic' });
    
```

La liste des **ATTRIBUTS** est placée entre les accolades {}. Lorsqu'un attribut n'est pas déclaré, celui-ci garde sa valeur par défaut.

Un objet peut être composé de plusieurs autres objets. La liste des attributs est rassemblée dans une **API Application Programming Interface**. Voir l'annexe 1 pour l'API de la jauge. Pour aller plus loin [<http://goo.gl/iJVOI6>]



Attribut	Type	Valeur par défaut
pointer	Objet	Voir Annexe 1
tickMajor	Objet	
tickMinor	Objet	

¹ Un **paradigme** est une représentation du monde, une manière de voir les choses

² Un **concept** est une représentation générale et abstraite de la réalité d'un objet

Dans l'exemple ci-dessous, seul l'attribut "value" a changé.



Value : _____



Value : _____

2.2. Construction de l'objet et affichage

Dans les pages web, les objets sont construits avec un langage de script.

Reprenons l'exemple précédent :

```
$("#gaugeContainer").jqxGauge({});
```

\$() ⇔ Rechercher dans la page HTML

• ⇔ Appliquer

⇔ id (identifiant)

Le code dans le script :

La fonction **\$()** recherche la balise identifiée par gaugeContainer dans la page Web.

```
$("#gaugeContainer").jqxGauge({
    // ... liste des attributs
});
```

La méthode de construction de l'objet "Gauge" jqxGauge est appliquée à cette balise avec les attributs placés entre accolades.

Ce code peut se traduire par :



Hé, l'élément avec l'ID gaugeContainer...

...j'ai créé une jauge. Affiche-la

```
$("#gaugeContainer").jqxGauge({
    // ... liste des attributs
});
```



... avec les valeurs des attributs par défaut ou celles présentes entre les accolades !

Le code dans le corps de la page HTML

Pour que le navigateur affiche cet objet, il faut lui préciser : "**sa position dans la page HTML**". C'est le rôle du code ci-dessous :

HTML

```
<div id="gaugeContainer"></div>
```

2.3. Comportement dynamique d'un objet : application d'une méthode

Exemple : Modification de la position de l'aiguille de la jauge après sa construction

```
$('#gaugeContainer').jqxGauge('value', 12);
```



2.4. Les évènements

Il est possible de contrôler le comportement dynamique des objets à partir des évènements susceptibles de se produire dans la page (clic de souris, clavier chargement de la page etc...)

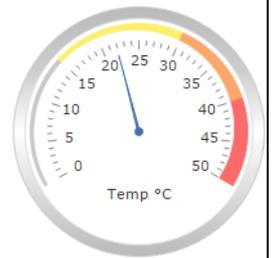
Exemple : Modification de la position de l'aiguille de la jauge avec un bouton



```
$('#setValueButton').jqxBUTTON({ width: 70, theme: 'classic' });
```



```
$('#setValueButton').click(function () {
    $('#gaugeContainer').jqxGauge('value', 22);
});
```



12°C 22°C

Le bouton s'affichera en plaçant `<input type="button" value="22°C" id=' setValueButton ' />` dans le corps de la page (body).

3. JQuery est une bibliothèque Javascript

Exemple : On souhaite remplacer un titre dans une page

```
<div id= "titre">Titre à remplacer</div>
```

Méthode Javascript pure et dure	Méthode JQuery
document.getElementById("titre").innerHTML = "Nouveau titre" ;	_____ ;

A retenir

1. Un objet possède un état décrit par des couples " _____ ". Valeur dépend du type de l'attribut.
2. Pour qu'un objet existe, il faut le créer avec un _____ !
3. Pour changer l'état d'un objet, on lui applique une _____ lorsqu'un _____ se produit !
4. jQuery est une _____ spécialisée dans la modification des pages web.
5. La fonction jQuery : _____ permet de sélectionner des éléments dans la page (CSS, HTML)

Annexe 1 : Documentation de l'objet "Gauge" (JQWidget)

Properties

Name	Type	Default
width	Number	350
height	Number	350
radius	Number	'50%'
startAngle	Number	30
endAngle	Number	270
value	Number	0
min	Number	0
max	Number	220
colorScheme	String	'scheme01'
disabled	Boolean	false
ticksDistance	Number	'20%"
animationDuration	Number	400
easing	String	linear
showRanges	Boolean	true
ranges	Array	[]
style	Object	{ fill: '#ffffff', stroke: '#ffffff' }
ticksMinor	Object	{ size: '10%', interval: 5, style: { stroke: '#898989'}, visible: true }
ticksMajor	Object	{ size: '10%', interval: 5, style: { stroke: '#898989'}, visible: true }
border	Object	{ size: '10%', style: { stroke: '#cccccc'}, visible: true, showGradient: true }
caption	Object	{ value: "", position: 'bottom', offset: [0, 0], visible: true }
cap	Object	{ size: '4%', style: { fill: 'theme-specific-color', stroke: 'theme-specific-color' }, visible: true }
pointer	Object	{ pointerType: 'default', style: { fill: 'theme-specific-color', stroke: 'theme-specific-color' }, length: '70%', width: '2%', visible: true }
labels	Object	{ distance: '38%', position: 'none', interval: 20, offset: [0, -10], visible: true, formatValue: function (value) { return value; } }

Events

Name	Arguments
valueChanging	Object
valueChanged	Object

Methods

Name	Arguments	Return Value
disable	None	None
enable	None	None

Annexe 2 : Mise en œuvre de l'objet "Gauge" (JQWidget) avec un script JQuery

```

<!DOCTYPE html>

<html>

<head>
  <title>jqxGauge Basic Demo</title>
  <link rel="stylesheet" href="jqwidgets/styles/jqx.base.css" type="text/css" />
  <script type="text/javascript" src="jqwidgets/scripts/jquery-1.7.2.min.js"></script>
  <script type="text/javascript" src="jqwidgets/jqxcore.js"></script>
  <script type="text/javascript" src="jqwidgets/jqxgauge.js"></script>
  <script type="text/javascript" src="jqwidgets/jqxbuttons.js"></script>

<script type="text/javascript">
  $(document).ready(function () {
    $('#gaugeContainer').jqxGauge({
      ranges: [{ startValue: 0, endValue: 15, style: { fill: '#C9C9C9', stroke: '#C9C9C9' }, endWidth: 3, startWidth: 1 },
              { startValue: 15, endValue: 30, style: { fill: '#FCF06A', stroke: '#FCF06A' }, endWidth: 6, startWidth: 3 },
              { startValue: 30, endValue: 40, style: { fill: '#FCA76A', stroke: '#FCA76A' }, endWidth: 9, startWidth: 6 },
              { startValue: 40, endValue: 50, style: { fill: '#FC6A6A', stroke: '#FC6A6A' }, endWidth: 12, startWidth: 9}],
      ticksMinor: { interval: 1, size: '5%' },
      ticksMajor: { interval: 5, size: '9%' },
      colorScheme: 'scheme01',
      labels: {interval:5},
      value: 22,
      max: 50,
      caption: {value:"Temp °C"},
      width: 200,
    });
  });
</script>

</head>

<body >
  <div id="gaugeContainer"></div>
</body>

</html>

```

Syntaxe à respecter