

AUCUNE DOCUMENTATION PERMISE

- (20) 1. Écrire le fichier de validation RelaxNG (forme compacte) pour le fichier XML suivant qui contient des notices biographiques d'informaticiens et d'informaticiennes.

```
<informaticiens>
    <informaticienne id="hopper" connaît="dijkstra knuth">
        <identité nom="Hopper" prénom="Grace"/>
        <dates naissance="1906-12-09" décès="1992-01-01"/>
        <contribution>COBOL</contribution>
        <aphorisme>Le bateau est en sécurité dans le port. Mais ce n'est pas pour cela que les bateaux ont été construits. Prenez la mer et faites de nouvelles choses.</aphorisme>
    </informaticienne>
    <informaticien id="turing">
        <identité prénom="Alan" nom="Turing"/>
        <dates naissance="1912-06-23" décès="1954-06-07"/>
        <contribution>Calculabilité</contribution>
        <contribution>Cryptanalyse d'Enigma</contribution>
        <contribution>Test d'intelligence artificielle</contribution>
    </informaticien>
    <informaticien id="dijkstra" connaît="knuth">
        <identité prénom="Edsger" nom="Dijkstra"/>
        <dates naissance="1930-05-11" décès="2002-09-06"/>
        <contribution>Sémaphore & section critique</contribution>
        <aphorisme>Le développeur <strong>compétent</strong> est totalement conscient du peu de talent qu'il a. C'est pourquoi il approche la tâche de programmation plein d'humilité et évite les <emph>trucs</emph> et <emph>astuces</emph> comme la peste.</aphorisme>
    </informaticien>
    <informaticien id="knuth" connaît="dijkstra">
        <identité nom="Knuth" prénom="Donald"/>
        <dates naissance="1938-01-10"/>
        <aphorisme>Faites attention aux bogues dans ce code ; je n'ai fait que démontrer qu'il était correct, je ne l'ai pas essayé.</aphorisme>
        <contribution>The Art of Computer Programming</contribution>
        <contribution>TeX</contribution>
        <contribution>Metafont</contribution>
    </informaticien>
</informaticiens>
```

- (10) 2. Identifiez et expliquez 4 points importants pour l'amélioration de l'ergonomie d'un site web.
- (10) 3. Pour chacune des questions suivantes, indiquez le numéro de l'énoncé vrai:
- A. Dans le contexte de l'interaction dans une page web, une action prise par un usager est représentée par:
 - (1) un objet
 - (2) un événement
 - (3) une balise
 - (4) un navigateur
 - B. Quelles sont les recommandations pour améliorer l'accessibilité d'un site web:
 - (1) utiliser la vidéo et le son aussi souvent que possible
 - (2) ajouter une description texte des fichiers audio et vidéo qui apparaissent sur vos pages web
 - (3) ne jamais utiliser d'audio ou de vidéo
 - (4) aucune de ces réponses
 - C. Quel attribut de l'élément **form** indique le nom et la localisation du script qui va traiter le contenu des champs du formulaire lorsqu'il est envoyé au serveur:
 - (1) **action**
 - (2) **process**
 - (3) **method**
 - (4) aucune de ces réponses
 - D. En HTML, comment indiquer le début d'un fragment d'une page appelé **top**?
 - (1) <div id="top" >
 - (2) <div bookmark="top" >
 - (3) <div id="#top" >
 - (4) <div href="top" >

(30) 4. HTML, CSS et SVG

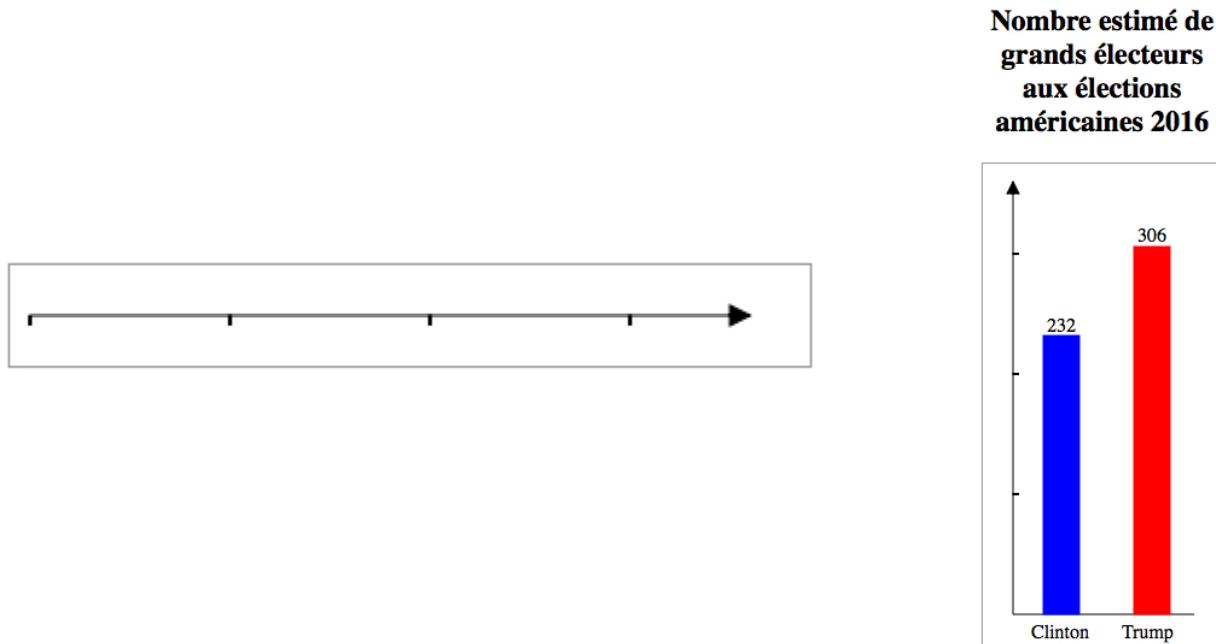


Figure 1: Un axe horizontal et un graphique en barre

(10) (a) Compléter l'élément SVG suivant et les déclarations CSS correspondantes

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      width="400" height="50" id="axe">

</svg>
```

pour reproduire la partie gauche de la figure 1 avec un élément `path` qui trace une ligne horizontale partant de (0,0) jusqu'à (350,0) avec des petites lignes verticales à tous les 100 sur l'axe des x ainsi qu'un petit triangle (10 de hauteur et 10 de largeur) à la fin de la ligne. L'origine de la ligne devrait apparaître au point (10,25).

(20) (b) En utilisant l'élément SVG défini en (a) et en considérant qu'il a été bien défini, créer la page HTML de la partie droite de la figure 1. Donner le code HTML contenant un élément SVG pour définir le graphique ainsi que le fichier CSS. La largeur du cadre du graphique est de 200 pixels et sa hauteur de 400 pixels. Les rectangles ont une largeur de 30 pixels et leur hauteur est donnée par le nombre de délégués obtenus par chaque candidat, tel qu'indiqué au-dessus de chaque barre. La barre correspondant à Clinton est bleue alors que celle correspondant à Trump est rouge.

Indice: Créer les barres à partir de l'origine (0,0) et ensuite effectuer une translation dans la surface de dessin.

- (30) 5. Soit le code javascript suivant (fichier `elections.js`):

```
// données tirées des résultats des élections générales québécoises entre 1960 et 1981
// source:https://fr.wikipedia.org/wiki/Liste_des_élections_générales_québécoises
data = [
    {"annee":1960, "lib":{"pct":51,"sieges":51}, "un":{"pct":47,"sieges":43}},
    {"annee":1962, "lib":{"pct":63,"sieges":56}, "un":{"pct":42,"sieges":31}},
    {"annee":1966, "lib":{"pct":47,"sieges":50}, "un":{"pct":41,"sieges":56}},
    {"annee":1970, "lib":{"pct":45,"sieges":72}, "pq":{"pct":23,"sieges":7},
     "un":{"pct":20,sieges:17}},
    {"annee":1973, "lib":{"pct":55,"sieges":102}, "pq":{"pct":30,"sieges":6},
     "un":{"pct":5,sieges:0}},
    {"annee":1976, "lib":{"pct":34,"sieges":26} , "pq":{"pct":41,"sieges":71},
     "un":{"pct":18,sieges:11}},
    {"annee":1981, "lib":{"pct":46,"sieges":42} , "pq":{"pct":49,"sieges":80},
     "un":{"pct":4,sieges:0}}
];
var partis=["lib","un","pq"];
var nomsPartis=["Libéraux","Union Nationale","Parti Québécois"]

function extraire(parti,type){
    ### à compléter (1)
}

$(document).ready(function() {
    ### à compléter (2)
})
```

- (10) (a) Écrire le code pour remplacer la section `### à compléter (1)` pour que cette fonction lorsqu'on l'appelle en lui donnant comme paramètre un parti ("lib", "un" ou "pq") et un type de données ("pct" ou "sieges") retourne une liste d'objets de la forme `{"annee":..., "valeur":...}` correspondant aux données de ce type pour toutes les années. La valeur "NA" est donnée lorsqu'il n'y a pas de donnée correspondante.

Par exemple, l'appel `extraire("pq", "sieges")` retournera:

```
[{"annee":1960,"valeur":"NA"}, {"annee":1962,"valeur":"NA"}, {"annee":1966,"valeur":"NA"}, {"annee":1970,"valeur":7}, {"annee":1973,"valeur":6}, {"annee":1976,"valeur":71}, {"annee":1981,"valeur":80}]
```

Résultats des élections québécoises de 1960 à 1981

Parti		1960	1962	1966	1970	1973	1976	1981
Libéraux	sièges	51	56	50	72	102	26	42
	%	51	63	47	45	55	34	46
Union Nationale	sièges	43	31	56	17		11	
	%	47	42	41	20	5	18	4
Parti Québécois	sièges				7	6	71	80
	%				23	30	41	49

Figure 2: Résultats des élections au Québec

- (20) (b) Reproduire le résultat donné dans la figure 2. Donnez le code HTML, le CSS ainsi que le code javascript qui remplace la section `###` à compléter (2) ci-dessus. Il est permis de définir de nouvelles fonctions pour simplifier votre code.

Cet examen comporte 6 questions pour un total de 100 points.

Bonne chance

Rappels RelaxNG compact

id = pattern

Sortes de *patterns*

element nom { pattern }
attribute nom { pattern }
mixed { pattern }

2 *patterns* combinés avec , ou & ou |
pattern suivi de ? ou * ou +

id

text

Rappels HTML

Structure d'une page

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8"/>
  <link href="..." type="text/css"/>
  <title>...</title>
</head>
<body>
  ...
</body>
</html>
```

Éléments du body

<i>sub, sup</i>	indice, exposant
<i>ul</i>	liste non-ordonnée
<i>ol</i>	liste ordonnée
<i>li</i>	élément de liste
<i>div, span</i>	regroupements d'éléments
<i>a href="..."</i>	lien entre documents
<i>img src="..." alt="..."</i>	image
<i>table</i>	tableau
<i>caption</i>	légende d'un tableau
<i>tr</i>	ligne d'un tableau
<i>th</i>	case d'entête de tableau
<i>td</i>	case de tableau

Éléments de division HTML5

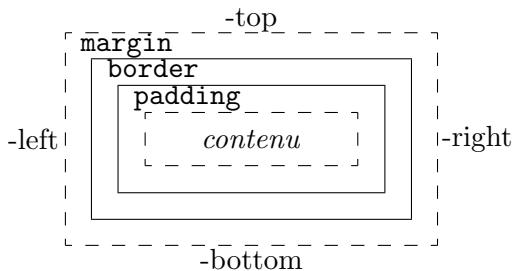
<i>header</i>	entête de toute la page
<i>nav</i>	navigation principale
<i>aside</i>	information auxiliaire
<i>section</i>	contenu de la page
<i>article</i>	unité d'information autonome
<i>footer</i>	pied de page

Rappels CSS

Sélecteurs

<i>nom</i>	nom d'un élément
<i>.classe</i>	nom d'une classe
<i>#id</i>	identificateur d'un élément
<i>:pseudo-classe</i>	<i>:link, :hover, :active</i>

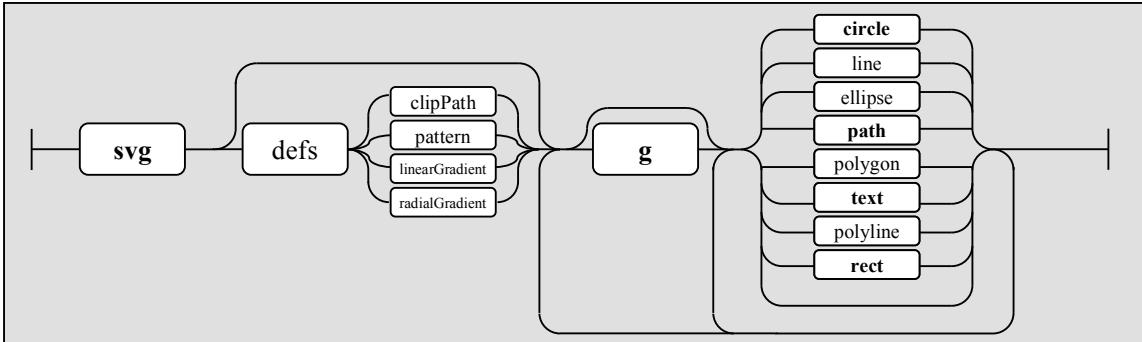
Modèle de boîte



Principales propriétés

<i>background-color</i>	couleur
<i>background-image</i>	<i>url(image)</i>
<i>background-repeat</i>	<i>repeat-x repeat-y no-repeat</i>
<i>background-position</i>	<i>left top center bottom right</i>
<i>border-width</i>	<i>longueur auto...</i>
<i>border-style</i>	<i>solid double...</i>
<i>border-width</i>	<i>thin thick longueur...</i>
<i>color</i>	spécification de la couleur
<i>display</i>	<i>inline, block</i>
<i>font-family</i>	<i>serif sans-serif ...</i>
<i>font-weight</i>	<i>normal bold ...</i>
<i>font-style</i>	<i>normal italic ...</i>
<i>height</i>	hauteur du bloc
<i>margin</i>	<i>longueur</i>
<i>padding</i>	<i>longueur</i>
<i>position</i>	<i>relative absolute fixed</i>
<i>text-align</i>	<i>left right center justify</i>
<i>top</i>	position en haut à gauche
<i>width</i>	<i>longueur</i>

SVG



Positioning :

Positioning	svg	g	rect	text	line	path	polygon	polyline	circle	ellipse	style	attr
transform	"translate(x,y)"		●	●	●	●	●	●	●	●	●	●
x,y	number			●	●							●
x1,x2,y1,y2	number					●						●
d	(special)						●					●
points	"x,y x,y x,y"							●	●			●
cx,cy	number									●	●	●

Sizing:

Sizing		svg	g	rect	text	line	path	polygon	polyline	circle	ellipse	style	attr
transform	"scale(k)"		●	●	●	●	●	●	●	●	●	●	●
width, height	number		●		●								●
r	number									●			●
rx, ry	number										●		●

Colors :

For text, fill refers to the color of the letters, stroke to that of the outline of the letters. If a pattern or a gradient is defined, it can be used with fill with its id.

Lines :

Line(s) :	svg	g	rect	text	line	path	polygon	polyline	circle	ellipse	style	attr
stroke-width	number			●	●	●	●	●	●	●	●	●
stroke-dasharray	numbers (separated by commas, blanks)*			●	●	●	●	●	●	●	●	●
stroke-linecap	butt round square				●	●	●	●	●			●
stroke-linejoin	miter round bevel					●	●	●	●			●
stroke-miterlimit	number					●	●	●	●			●

the numbers represent dash lengths and gaps.

Text :

Property	Value	Description	Style	Attr
text-anchor	start middle end	Determine if the text is aligned to the left, centered or aligned to the right.		●
writing-mode	tb	If specified, text is written vertically		●
glyph-orientation-vertical	0 1	If 0, and writing-mode is set to tb, letters are horizontal (stacked), else if set to 1 when text mode is set to tb, the text will be rotated (written from top to bottom).		●
textLength	Number	If specified, the text will be adjusted to fit the specified length.		●
lengthAdjust	spacing spacingAndGlyphs	Which strategy is used to adjust the length?		●
textPath	xlink:href="#id)	Writes the text along a path (which should be in the defs and given an id attribute)		

Special :

Selectors	Basics	Hierarchy	Core	AJAX	Shorthand Methods
	#id	ancestor	jQuery function	Low-Level Interface	\$.load(url [, data] [, fn(responseText, status)])
	element	descendant	\$.jQuery(selector [, context] element elementArray)	jqXHR .jQuery.ajax(options, [settings])	jqXHR .jQuery.get(url [, data] [, fn(data, status, jqXHR)])
	.class	parent >	\$.jQuery(html [, owner] html, props)	jqXHR .jQueryObject(accept, config)	jqXHR .jQuery.getJSON(url [, data] [, fn(data, status)])
	.class.class	child	\$.jQuery(fn[, index])	fn beforeSend(jqXHR, config)	jqXHR .jQuery.getScript(url [, fn(data, status)])
	*	prev +	def .when(deferreds)	fn complete(jqXHR, status)	jqXHR .jQuery.post(url [, data] [, fn(data, status)], type)
	selector1,	next	fn .when(doneCallbacks)	str contentType	
	selector2	prev ~	\$.when(deferreds)	map converters	
		siblings	\$.when(sub())	obj str data	
			\$.holdReady(hold)	bool global = true	
Basic Filters	Content	Filters		bool ifModified = false	
	:first			fn jsonpCallback	
	:last			bool processData = true	
	:not(selector):empty			map statusCode	
	:even			bool traditional	
	:odd			str url = curr. page	
	:eq(index)	Visibility		fn xhr	
	:gt(index)			str dataType = {xml, json, script, html}	
	:lt(index)	Filters		fn error(jqXHR, status, errorThrown)	
	:header	hidden		fn success(data, status, jqXHR)	
Child Filters	:animated	visible			
	:focus				
	:nth-child(expr)				
	:first-child	[attribute]=value]			
	:last-child	[attribute]=value]			
	:only-child	[attribute^=value]			
		[attribute*=value]			
		[attribute]=value]			
		[attribute~=value]			
		[attribute]			
Forms		Attributes			
		Attributes			
Traversing	Form Filters		CSS	Events	
	:input	:enabled	str .css(name)	Page Load	\$.ready(fn())
	:text	:disabled	\$.css(name, val map name, fn(index, val))		
	:password	:checked			
	:radio	:selected			
	:checkbox				
	:submit				
	:image				
	:reset				
	:button				
Filtering	:file				
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
Filtering					
			</		