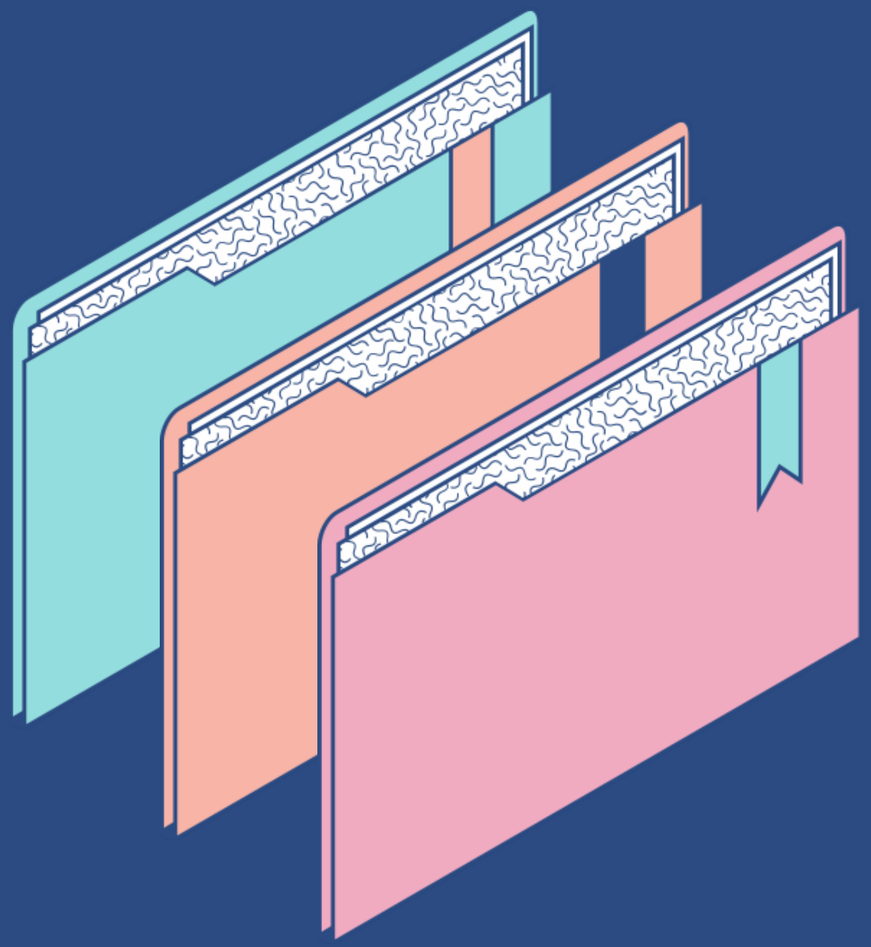
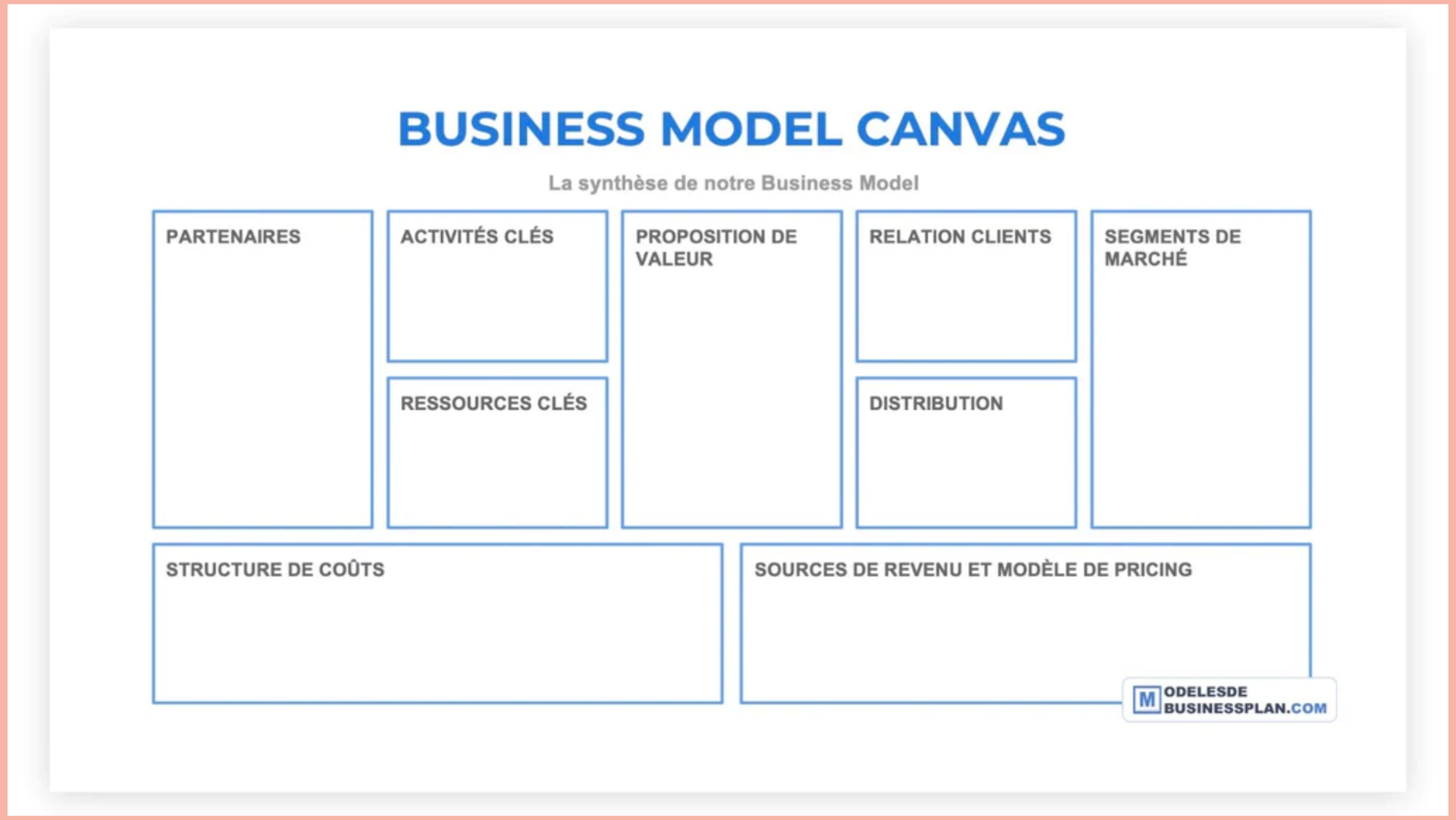
Javascript


JavaScript
<script>
if(age > 19){
alert("Adult");
} else{
alert("Teenager");
}
</script>

SOMMAIRE



Canvas HTML


BUSINESS MODEL CANVAS
La synthèse de notre Business Model
PARTENAIRES
ACTIVITÉS CLÉS
RESSOURCES CLÉS
PROPOSITION DE VALEUR
RELATION CLIENTS
DISTRIBUTION
SEGMENTS DE MARCHÉ
STRUCTURE DE COÛTS
SOURCES DE REVENU ET MODÈLE DE PRICING
MODELESDE BUSINESSPLAN.COM

Présentation Canvas HTML

L'élément HTML Canvas est un conteneur, au sein duquel nous pourrons dessiner des graphiques en JS.

La taille de l'élément canvas se définit avec les attributs width et height.

Desiner dans un canvas

Pour pouvoir dessiner au sein du canvas, il faut :

- Accéder à l'élément canvas via JS
- Accéder au contexte de rendu du canvas
- Utiliser les propriétés et méthodes pour effectuer le dessin

```
let canvas = document.getElementById('c1');

let ctx = canvas.getContext('2d');
```

Dessiner un rectangle vide

strokeRect(a,b,c,d) : permet de dessiner un rectangle vide où :
a la marge gauche; b la marge supérieure; c la largeur; d la hauteur

```
let canvas = document.getElementById('c1');
let ctx = canvas.getContext('2d');

ctx.strokeStyle = '#4444CC'; //Nuance de
bleu
ctx.strokeRect(50, 25, 200, 100);
```

Dessiner un rectangle plein

fillRect(a,b,c,d) : permet de dessiner un rectangle plein où :
a la marge gauche; b la marge supérieure; c la largeur; d la hauteur

```
let canvas = document.getElementById('c1');
let ctx = canvas.getContext('2d');

ctx.fillStyle = '#4444CC'; //Nuance de bleu
ctx.fillRect(50, 25, 200, 100);
```

Effacer une zone rectangulaire

clearRect(a,b,c,d) : permet de supprimer un rectangle où :
a la marge gauche; b la marge supérieure; c la largeur; d la hauteur

```
ctx.strokeRect(350, 175, 200, 100);

ctx.clearRect(150, 75, 300, 150);
```

Dessiner une ligne

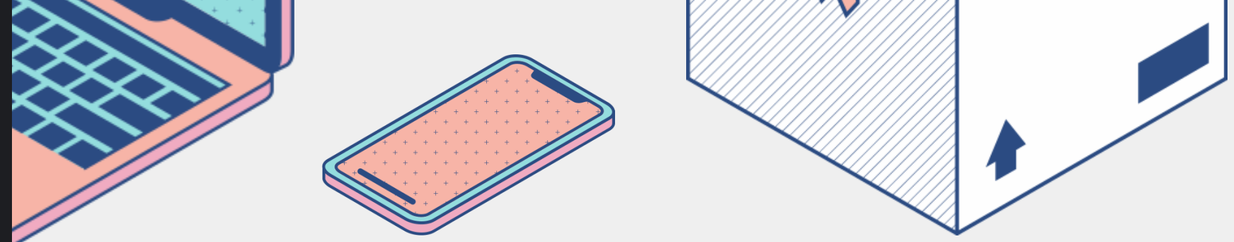
beginPath() :
Permet de démarrer le tracé
lineTo(x,y) :
définit une ligne finissant en xy,
démarrant à dernière position tracée
moveTo(x,y) :
définit le point de départ de la ligne
stroke() :
Valide le tracé de la ligne

```
let canvas = document.getElementById('c1');
let ctx = canvas.getContext('2d');

ctx.beginPath();
ctx.moveTo(50, 25);
ctx.lineTo(250, 125);
ctx.strokeStyle= '#4488EE'; //Nuance de
bleu
ctx.lineWidth= 3;
ctx.stroke();
```

Arcs de cercle

arc(a,b,c,d,e,f) : Permet de dessiner des arcs de cercle où :
a représente le décalage entre le centre du cercle et la gauche du canvas
b le décalage du point central et le bord supérieur
c la taille du rayon
d l'angle de départ en radians
e l'angle de fin en radians
f (facultatif) (bool) dessiné dans le sens des aiguilles d'une montre (false) ou inverse (true)



```
let canvas = document.getElementById('c1');
let ctx = canvas.getContext('2d');

//Arc de cercle vert
ctx.beginPath();
ctx.lineWidth = '5';
ctx.strokeStyle = '#4C8';
ctx.arc(50,50,35,0.8*Math.PI, 2*Math.PI);
ctx.closePath();
ctx.stroke();

//Cercle complet violet
ctx.beginPath();
ctx.lineWidth = '5';
ctx.fillStyle = '#A4A';
ctx.arc(150,85,40,0,2*Math.PI);
ctx.fill();

//Arc de cercle bleu
ctx.beginPath();
ctx.lineWidth = '5';
ctx.strokeStyle = '#48C';
ctx.arc(250,50,35,0.2*Math.PI, Math.PI, true);
ctx.closePath();
ctx.stroke();
```

Arcs de cercle

Dégradés linéaires

createLinearGradient(a,b,c,d) : permet de créer un dégradé linéaire où :
a est l'écart entre le point de départ du dégradé et le bord gauche du canvas
b est l'écart entre le point de départ du dégradé et le haut du canvas
c est l'écart entre le point de fin du dégradé et la gauche du canvas
d est l'écart entre le point de fin du dégradé et le haut du canvas

addColorStop(a,b) : Permet de créer des transitions entre le couleurs où :
a est le point d'arrêt (compris entre 0 et 1)
b est la couleur

Dégradés linéaires

```
let canvas = document.getElementById('c1');
let ctx = canvas.getContext('2d');

let lineaire = ctx.createLinearGradient(25, 25, 100, 25);
lineaire.addColorStop(0,'#4C8'); //Vert
lineaire.addColorStop(0.5, '#48C'); //Bleu
lineaire.addColorStop(1, '#A4A'); //Violet

ctx.fillStyle = lineaire;
ctx.fillRect(25, 25, 75, 100);

let lineaire2 = ctx.createLinearGradient(150, 25, 275, 125);
lineaire2.addColorStop(0,'#DD4'); //Jaune
lineaire2.addColorStop(1, '#D44'); //Rouge

ctx.beginPath();
ctx.moveTo(150, 25);
ctx.lineTo(150, 125);
ctx.lineTo(275, 125);
ctx.fillStyle = lineaire2;
ctx.fill();
```

Texte

fillText(a,b,c) : Permet d'ajouter du texte plein sur un canvas où :
a est le texte, et b;c sont les coordonnées

strokeText(a,b,c) : même fopnctionnement que fillText mais pour des textes creux.

Nous pourrons customiser notre texte à l'aide des propriétés font, textAlign, textBaseline, direct

Texte

```
let canvas = document.getElementById('c1');
let ctx = canvas.getContext('2d');


ctx.font = 'bold 20px Verdana, Arial, serif';

ctx.strokeStyle = '#48B';
ctx.strokeText('Texte creux', 25, 50);

ctx.font = 'bold 20px Verdana, Arial, serif';
ctx.fillStyle = '#48B';
ctx.textAlign = 'center'; //Le milieu du texte
sera à 150
ctx.fillText('Texte plein', 150, 100);
```

Texte creux

Texte plein

Image

drawImage(a,b,c) : dessine un objet image, où :

a est la référence à l'image

b est la position de l'image sur le canvas

c (facultatif) est la largeur et hauteur de l'image à insérer

```
let canvas = document.getElementById('c1');
let ctx = canvas.getContext('2d');
let image = document.getElementById('sunset');

image.addEventListener('load', function(){
    ctx.drawImage(image, 75, 25, 150, 100);
}, false);
```

Exercice d'application : Dessiner un Canvas

Choisissez un animal de votre choix, et dessinez le au sein d'un canvas.
Si vous n'êtes pas à l'aise avec le canvas, prenez un animal très simple, et dessinez le de façon très grossière.

Tips : Les dessins suivants sont uniquement à titre d'exemple !
Commencez par le dessiner à la main et reproduisez le sur Canvas ensuite.