



ACADEMY OF DIGITAL ARTS EGYPT






START
YOUR TECH JOURNEY
WITH ADA





Agenda

- Box Model
 - Layouts
 - Flexbox
- 

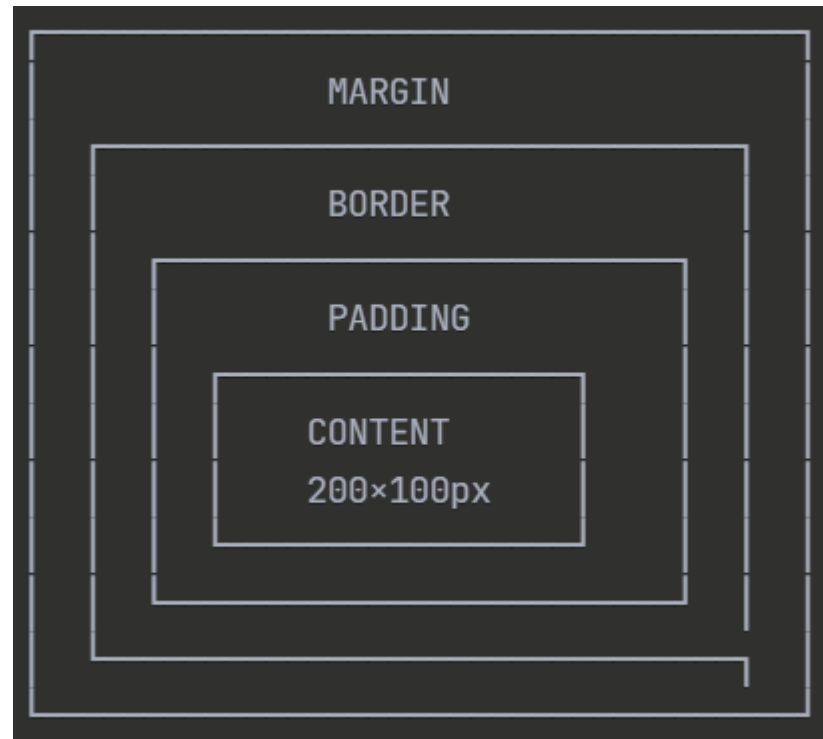
CSS Box Model Deep Dive

CSS Box Model Deep Dive: The Four Components

```
.box {  
  width: 200px;           /* Content area */  
  padding: 20px;          /* Space inside the border */  
  border: 2px solid #333; /* Border around padding */  
  margin: 10px;           /* Space outside the border */  
}
```

CSS Box Model Deep Dive

CSS Box Model Deep Dive: Visual Representation



CSS Box Model Deep Dive

CSS Box Model Deep Dive: Box-Sizing Property

```
/* Default behavior */  
.default-box {  
  box-sizing: content-box;  
  width: 200px;  
  padding: 20px;  
  border: 2px solid #333;  
  /* Total width = 200 + 40 + 4 = 244px */  
}
```

```
/* Modern approach */  
.modern-box {  
  box-sizing: border-box;  
  width: 200px;  
  padding: 20px;  
  border: 2px solid #333;  
  /* Total width = 200px (includes padding and border) */  
}
```



CSS Box Model Deep Dive

CSS Box Model Deep Dive: Universal Box-Sizing Reset

```
*, *::before, *::after {  
  box-sizing: border-box;  
}
```



Margin and Padding

Individual Properties:

```
.spacing-example {  
  margin-top: 10px;  
  margin-right: 15px;  
  margin-bottom: 20px;  
  margin-left: 25px;  
  
  padding-top: 5px;  
  padding-right: 10px;  
  padding-bottom: 15px;  
  padding-left: 20px;  
}
```


Margin and Padding

Shorthand Properties:

```
.shorthand-examples {  
  /* All sides equal */  
  margin: 20px;  
  padding: 15px;  
  
  /* Vertical | Horizontal */  
  margin: 10px 20px;  
  padding: 15px 25px;  
  
  /* Top | Horizontal | Bottom */  
  margin: 10px 20px 30px;  
  padding: 5px 15px 25px;  
  
  /* Top | Right | Bottom | Left (clockwise) */  
  margin: 10px 15px 20px 25px;  
  padding: 5px 10px 15px 20px;  
}
```

Margin and Padding

Margin Collapse and Auto Centering:

```
.centered-element {  
  width: 800px;  
  max-width: 90%;  
  margin: 0 auto; /* Horizontal centering */  
}  
  
.margin-collapse-demo {  
  margin-bottom: 30px;  
}  
  
.margin-collapse-demo + .margin-collapse-demo {  
  margin-top: 20px;  
  /* Only 30px space between elements, not 50px */  
}
```

Border Properties

Border Components:

```
.border-examples {  
    /* Individual properties */  
    border-width: 2px;  
    border-style: solid;  
    border-color: #007bff;  
  
    /* Shorthand */  
    border: 2px solid #007bff;  
  
    /* Individual sides */  
    border-top: 3px solid red;  
    border-right: 1px dashed blue;  
    border-bottom: 2px dotted green;  
    border-left: 4px solid orange;  
}
```

Border Properties

Border Styles and Effects:

```
.border-styles {  
  border-style: solid;      /* ————— */  
  border-style: dashed;     /* - - - - - */  
  border-style: dotted;     /* ..... */  
  border-style: double;     /* ===== */  
  border-style: groove;     /* 3D effect */  
  border-style: ridge;      /* 3D effect */  
  border-style: inset;      /* 3D effect */  
  border-style: outset;     /* 3D effect */  
}
```

```
.border-radius {  
  border: 2px solid #333;  
  border-radius: 10px;      /* All corners */  
  border-radius: 10px 20px; /* Top-left/bottom-right | Top-right/bottom-left */  
  border-radius: 5px 10px 15px 20px; /* Individual corners */  
  
  /* Perfect circle (when width = height) */  
  border-radius: 50%;  
}
```

CSS Layouts Fundamentals

1- Block vs Inline vs Inline-Block

```
.block-element {  
  display: block;  
  /* Takes full width, new line before and after */  
  width: 200px;  
  height: 100px;  
  margin: 20px;  
  padding: 15px;  
}
```

```
.inline-element {  
  display: inline;  
  /* Only takes needed width, no line breaks */  
  /* Width and height ignored */  
  /* Only horizontal margin/padding effective */  
  margin: 20px 10px; /* Only horizontal margins work */  
  padding: 15px 10px; /* Vertical padding doesn't affect layout */  
}  
  
.inline-block-element {  
  display: inline-block;  
  /* Best of both: respects width/height, stays in line */  
  width: 200px;  
  height: 100px;  
  margin: 20px;  
  padding: 15px;  
}
```

CSS Layouts Fundamentals

2- None and Hidden

```
.hidden-element {  
    display: none; /* Completely removes from layout */  
}  
  
.invisible-element {  
    visibility: hidden; /* Hidden but space preserved */  
}
```


CSS Layouts Fundamentals

3- Position Property

```
.static-position {  
    position: static; /* Default - follows normal flow */  
}  
  
.relative-position {  
    position: relative;  
    top: 10px;      /* Moves 10px down from normal position */  
    left: 20px;     /* Moves 20px right from normal position */  
    /* Original space is preserved */  
}
```

```
.absolute-position {  
    position: absolute;  
    top: 50px;      /* 50px from nearest positioned ancestor */  
    right: 20px;    /* 20px from right edge of positioned ancestor */  
    /* Removed from normal flow */  
}  
  
.fixed-position {  
    position: fixed;  
    bottom: 20px;   /* 20px from bottom of viewport */  
    right: 20px;    /* 20px from right of viewport */  
    /* Fixed relative to viewport */  
}  
  
.sticky-position {  
    position: sticky;  
    top: 0;         /* Sticks to top when scrolling past */  
    /* Hybrid between relative and fixed */  
}
```

Flexbox Layout System

Creating a Flex Container

```
.flex-container {  
  display: flex;  
  /* All direct children become flex items */  
}  
  
.inline-flex-container {  
  display: inline-flex;  
  /* Flex container that behaves like inline-block */  
}
```

Flexbox Layout System

Flex Direction

```
.flex-row {  
  display: flex;  
  flex-direction: row;      /* Default: left to right */  
}  
  
.flex-row-reverse {  
  display: flex;  
  flex-direction: row-reverse; /* Right to left */  
}  
  
.flex-column {  
  display: flex;  
  flex-direction: column;    /* Top to bottom */  
}  
  
.flex-column-reverse {  
  display: flex;  
  flex-direction: column-reverse; /* Bottom to top */  
}
```

Flexbox Layout System

Flex Wrap

```
.flex-wrap {  
  display: flex;  
  flex-wrap: nowrap;    /* Default: items shrink to fit */  
  flex-wrap: wrap;      /* Items wrap to new lines */  
  flex-wrap: wrap-reverse; /* Items wrap in reverse order */  
}  
  
/* Shorthand for direction and wrap */  
.flex-flow {  
  display: flex;  
  flex-flow: row wrap; /* flex-direction flex-wrap */  
}
```

Flex Container Alignment

Justify Content

```
.justify-examples {  
  display: flex;  
  
  justify-content: flex-start;    /* Default: items at start */  
  justify-content: flex-end;      /* Items at end */  
  justify-content: center;        /* Items centered */  
  justify-content: space-between; /* Equal space between items */  
  justify-content: space-around;  /* Equal space around items */  
  justify-content: space-evenly;  /* Equal space everywhere */  
}
```

Flex Container Alignment

Align Items

```
.align-examples {  
  display: flex;  
  height: 200px; /* Height needed to see vertical alignment */  
  
  align-items: stretch; /* Default: items stretch to fill */  
  align-items: flex-start; /* Items align to top */  
  align-items: flex-end; /* Items align to bottom */  
  align-items: center; /* Items centered vertically */  
  align-items: baseline; /* Items align to text baseline */  
}
```


Flex Container Alignment

Perfect Centering

```
.perfect-center {  
  display: flex;  
  justify-content: center; /* Horizontal centering */  
  align-items: center;    /* Vertical centering */  
  min-height: 100vh;     /* Full viewport height */  
}
```

Flex Items Properties

Flex Grow, Shrink, and Basis

```
.flex-item-1 {  
  flex-grow: 1;    /* Takes 1 part of available space */  
  flex-shrink: 1;  /* Can shrink if needed */  
  flex-basis: 0;   /* Starting size before growing/shrinking */  
}  
  
.flex-item-2 {  
  flex-grow: 2;    /* Takes 2 parts of available space */  
  flex-shrink: 0;  /* Cannot shrink */  
  flex-basis: 200px; /* Starts at 200px */  
}  
  
/* Shorthand property */  
.flex-shorthand {  
  flex: 1;          /* flex: 1 1 0% */  
  flex: 0 0 200px; /* flex: grow shrink basis */  
  flex: none;       /* flex: 0 0 auto */  
}
```

Flex Items Properties

Individual Item Alignment

```
.special-alignment {  
  align-self: flex-start; /* Override container's align-items */  
  align-self: flex-end;  
  align-self: center;  
  align-self: stretch;  
  align-self: baseline;  
}
```

Flex Items Properties

Flex Order

```
.flex-order {  
  order: 0; /* Default order */  
  order: 1; /* Appears after order: 0 items */  
  order: -1; /* Appears before order: 0 items */  
}
```

THANK YOU

ADAEGY

