

# LAYOUT BLOCKS

Basic Components make up the general layout of the course.  
They are divided in 2 types, Blocking and Widgets.

## SECTION || BLOCK - CONTAINERS

### SECTION

A Section container.

Wraps components that make up a section of a challenge

### Props

#### type

1. White(default) **OR** Grey

#### collapse

1. Desktop(default) **OR** Tablet  
*\*Tablet forces Columns Inside Section To Collapse at Tablet Size*

### BLOCK

A Blocking Container

Wraps components that make up a section of a challenge

### Props

#### type

1. White **OR** Grey (default: Grey)
2. Wide (default: OFF *\*approx. 8% padding*)
3. Column-2 **OR** Column-3 (default: OFF)
4. Box (default: OFF *\*border, rounded*)

### Padding and Margin

**{Margin OR Padding} – {10,20,30,40,50 \*px} – {Top OR Bottom}**

#### For Example

*Padding-Top-10*

*Margin-Bottom-40*

*Margin-Bottom*

*Padding-Top*

*\*default 40px*

```
<Section type="Grey">
  <Block type="Column-2" collapse="Tablet">
    <Block type="White Wide">
      <h4 className="Padding-20">TFO Layout's</h4>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
    </Block>
    <Block type="Box">
      <h4>Welcome to TFO Layout's</h4>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
    </Block>
  </Block>
</Section>
```

#### Welcome to TFO Layout's

Lorem ipsum dolor sit amet consectetur  
adipisicing elit.

#### Welcome to TFO Layout's

Lorem ipsum dolor sit amet consectetur  
adipisicing elit.

# TEXT COMPONENTS

## TEXT || LISTS

### LISTS

#### Props

##### type

1. **large** \*default
2. **small**
3. **sub**
4. **alphabetic** a.
5. **numeric** 9.

##### list

1. **List Items**

\* Multiple items can be placed in array  
list = [{"item1"}, "item2"]

##### indent

1. **true**
2. **false** \*default

- un logiciel de traitement de texte un logiciel de correction
- un logiciel de correction grammaticale (p. ex., Antidote)
- un logiciel de traitement de texte un logiciel de correction
- un logiciel de correction grammaticale (p. ex., Antidote)
  - un logiciel de traitement de texte un logiciel de correction
  - un logiciel de correction grammaticale (p. ex., Antidote)
- a. un logiciel de traitement de texte un logiciel de correction
- b. un logiciel de correction grammaticale (p. ex., Antidote)
- 9. un logiciel de traitement de texte un logiciel de correction
- 10. un logiciel de correction grammaticale (p. ex., Antidote)

```
<List
  indent={true}
  type={"sub"}
  list=[
    `un logiciel de traitement de texte un logiciel de correction`,
    `un logiciel de correction grammaticale (p. ex., Antidote)`,
  ]
/>
```

### Common HTML Elements

#### Common HTML Elements

1. **p**
2. **span**
3. **b** \*Bold & **i** \*Italic
4. **h1** → **h6**
5. **ul/ol** \*When List Component is not sufficient

### Common Classes

#### Common HTML Elements

1. **"h1" → "h5"** (Heading hierarchy, AODA)
2. **italic & bold**
3. **clearfix** (Clear's floated items)

#### Nested JSX elements

In order for the code to compile, a JSX expression must have exactly one outermost element.

#### React Element Attributes

React DOM uses camelCase property naming convention instead of HTML attribute names.

#### React Element Styles

To apply inline styles, we use two sets of curly braces. Inline-styles are not written as plain strings, but as plain objects:

```

<br />
<hr />
```

```
<div className="Goodbye">
  <h1>Goodbye World!</h1>
</div>
```

```
<div>
  <h1 style={{color: "red"}}>Hello World!</h1>
</div>
```

# IMAGE BLOCKS

## IMAGES

### IMAGE

### Props

#### position

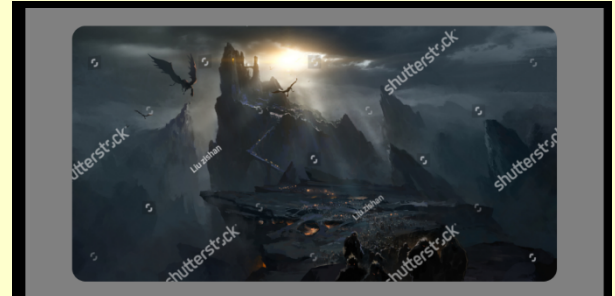
1. left
2. right
3. center \*default

#### image

1. Source / Link to Image indent

#### alt

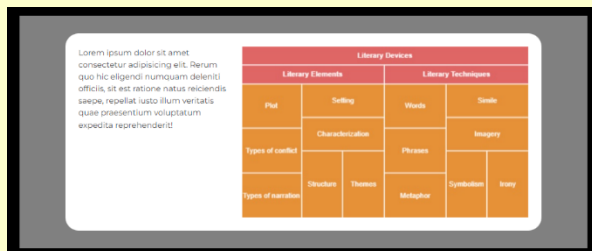
1. alt text for Screen reader



```
<Section type="Grey">
  <Image
    img="assets/Module1/img/01/05-01.png"
    alt="A series of 30 words of different sizes and colours"
    position="center"
  />
</Section>
```

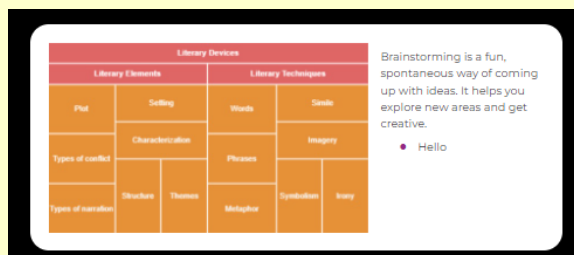
I

Anything placed alongside an Image must be wrapped by the Image Tags



```
<Section type="Grey">
  <Image
    img="assets/Module1/img/01/02-02.jpg"
    alt="A series of 30 words of different sizes and colours that are put
    together in a way that forms the shape of a cloud."
    position="right"
  >
  <p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Rerum quo
    hic eligendi numquam deleniti officiis, sit est ratione natus
    reiciendis saepe, repellat iusto illum veritatis quae praesentium
    voluptatum expedita reprehenderit!
  </p>
</Image>
</Section>
```

Can put any HTML element, as well as List Component



```
<Section type="Grey">
  <Image
    img="assets/Module1/img/01/02-02.jpg"
    alt="A series of 30 words of different sizes and colours"
    position="left"
  >
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
  <List type="numeric" list=["item 1", "item 2"] />
</Image>
</Section>
```

# PRE-BUILT

## BANNERS || BLOCKQUOTES

### BANNER

#### Props

##### heading

##### subheading

1. Content (Text)

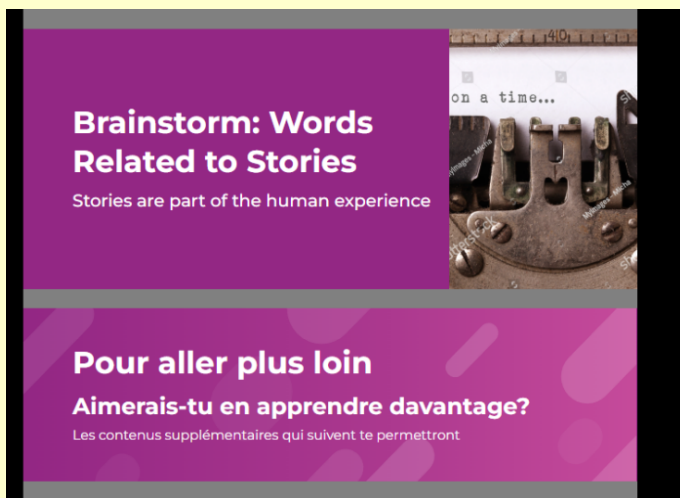
##### image

1. Source / Link to Image

##### type

1. activity
2. consolidation
3. learn-more

```
<Banner
  heading={"Brainstorm: Words Related to Stories"}
  subheading={"Stories are part of the human experience"}
  image={"assets/Module1/img/01/sample.png"}
/>
```



```
<Banner
  heading={"Pour aller plus loin"}
  subheading={"Aimerais-tu en apprendre davantage?"}
  type={"learn-more"}
  text={"Les contenus supplémentaires qui suivent te permettront"}
/>
```

### BLOCKQUOTE

#### Props

##### text

1. Content (Text)

##### type

1. Large OR Small (default:Large)  
\*Large = Blue  
\*Small = Purple

```
<Section>
  <Blockquote text="Sample" />
</Section>
```

How do the literary devices found in narration (e.g., plot, character, conflict, setting, etc.) help properly structure a story in order to allow for a deeper understanding?

How do the literary devices found in narration (e.g., plot, character, conflict, setting, etc.) help properly structure a story in order to allow for a deeper understanding?

```
<Section>
  <Blockquote text="Sample" type="small" />
</Section>
```

# PRE-BUILD - CONT

## LEARNING GOALS || SKILLS & HABITS || THEMATIC VOCABULARY

### LEARNING GOALS

#### Props

##### goals

1. list of goals \*array

#### Learning Goals for This Challenge

- I understand how a story is built and appropriate the different elements that are imperative to the analysis of a narrative
- I identify how an author is influenced by their surroundings and culture when writing a story.
- I enrich my vocabulary by learning the terms and words recommended in the module.
- I show my knowledge of the vocabulary by applying it to a story.

#### Learning Skills and Work Habits for This Challenge

**Independent work**  
I work on the required assignments without reminders to submit the work on time.

### LEARNING SKILLS & HABITS

#### Props

##### skills

1. list of skills \*array

##### habits

1. list of habits \*array

```
<Section>
  <LearningGoals goals={['Goal 1`, `Goal 2`, `Goal 3`, `Goal 4`]} />
  <LearningSkillsHabits skills={['Skill`']} habits={['Habit`']} />
</Section>
```

### THEMATIC VOCABULARY

#### Props

##### text

1. Line of text *\*before list of words*

##### list

1. list of words to list \*array

#### Thematic Vocabulary for This Challenge

You will discover the following words or expressions during your challenge. Make sure to apply reading strategies to develop an understanding of these words.

literary devices	protagonist
character	
plot	
novel	
setting	

```
<Thematic_Vocabulary
  title={`Thematic Vocabulary for This Challenge`}
  text={
    `You will discover the following words or expressions during your challenge.
    Make sure to apply reading strategies to develop an understanding of these words.`
  }
  list=[
    `literary devices`,
    `character`,
    `plot`,
    `novel`,
    `setting`,
    `protagonist`,
  ]
/>
```

After 5 words list  
breaks into 2 columns  
on desktop

# Component Props

## Basic Interactive Components (Widgets)

### Accordion

- title: Title of each accordion
- text: If there is only one paragraph, then data is a string (eg. text = "text")  
text: If there are more than one paragraphs, then data is in an array (eg. text = ['text', 'text2'])
- img: If there is no image, no img props needed.
- img: if there is an image, the data is in the following format:
  - src: Path to assets
  - alt: Image alt text
  - size: The value must be defined by WPAs 'landscape' or 'portrait'  
\*\*\* The size value only needs to be indicated in Accordion.

### Tabs

- title: Title of each Tab
- text: If there is only one paragraph, then data is a string
- text: If there are more than one paragraph, then data is in an array.
- img: if there is an image, the data is in the following format:
  - src: Path to assets
  - alt: Image alt text

### Timeline Tabs

If tab is timeline, props in <Tab> requires type="timeline".

For example, <Tab type="timeline">

- title: Title of each Timeline Tab
- list: If there is a single bullet list, then data is a string. (eg. list="list")
- list: If there are more than one bullet list, then data is in an array (eg. list=['list1', 'list2'])
- text: If there is no text paragraph, no text props needed.
- text: If there is one paragraph, then data is a string. If there is more than one paragraph, the data is in an array.
- img: If there is no image, no img props needed.
- img: if there is an image, the data is in the following format:
  - src: Path to assets
  - alt: Image alt text

### Carousel

Props type in <Carousel> will differentiate the design and format.

Photo Carousel: <Carousel type="photo">

Text Carousel: <Carousel type="text">

### Photo Carousel

- text: There is going to be only single paragraph, and data always be a string.
- img: if there is an image, the data is in the following format:
  - src: Path to assets
  - alt: Image alt text

## Text Carousel (Stage/Steps)

- heading: Title of each carousel
- subheading: Sub-title of each carousel
- text: either one paragraph or more
  - one paragraph: a string (eg. text="text")
  - more than one paragraph: an array (eg. text=["text1", "text2"])
- img: if no image required, no image props needed.
- img: if there is an image, the data is in the following format:
  - src: Path to assets
  - alt: Image alt text

## Audio

- audioData: Name of audio file
- button: "ACCÉDER À LA TRANSCRIPTION" --- This is a boilerplate text.

\*\*\* The audio and transcript file must have the same name. (eg. audio.mp3, audio.html)

## Video

- Using TFO's Idello player

## Interactive Components (ILOs)

Tooltip

Flip Cards

Image with Hotspots

Multiple Choice: Radio buttons

Multiple Choice: Checkbox

Multiple Choice: True/False

Drag and Drop: Flowchart

Drag and Drop: Matching

Drag and Drop: Sorting