

# Forms

---



**Liam McLennan**

CTO

@liammclennan [withouttheloop.com](http://withouttheloop.com)



# Overview



**Form elements**

**User input**

**Form libraries**

**Validation**

**Accessing DOM elements**



# Form Elements



**Just like HTML**

**Preserve React's rendering semantics**



```
<input type="text" value="react" />
```

Text Input



# HTML

```
<textarea>  
  react  
</textarea>
```

# React

```
<textarea value="react"/>
```



# HTML

```
<select>
  <option value="saturday">
    Saturday
  </option>
  <option value="sunday"
    selected>
    Sunday
  </option>
</select>
```

# React

```
<select value="sunday">
  <option
    value="saturday">
    Saturday
  </option>
  <option
    value="sunday">
    Sunday
  </option>
</select>
```

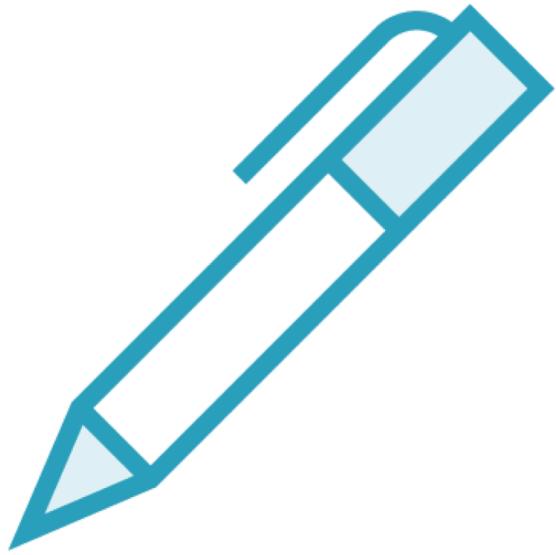


```
<select value="sunday">  
    <option value="saturday">Saturday</option>  
    <option value="sunday">Sunday</option>  
</select>
```

Select



# Allowing User Input



**Form elements are read-only**  
**Component state supports editing**

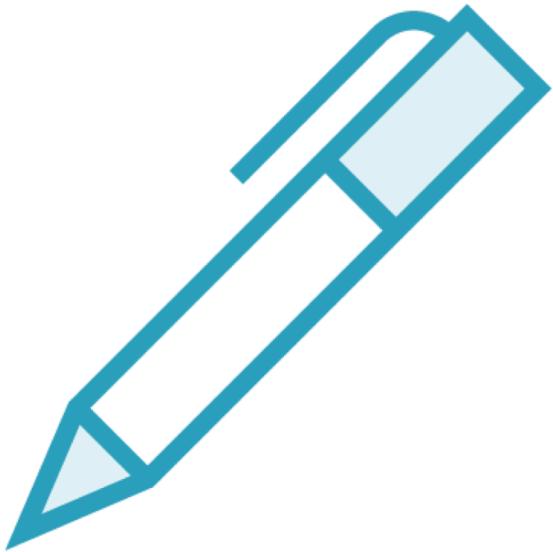


# A Read-only Form

```
class Identity extends React.Component {  
  render() {  
    return (  
      <form>  
        <input type="text" value="" placeholder="First name" />  
        <input type="text" value="" placeholder="Last name" />  
      </form>);  
    }  
}
```



# Allowing User Input



**Add state to the component**

**Bind inputs to the component state**

**Use onChange handler to update state**



# Form Libraries



**Forms can be time consuming**

**Many form libraries are available**

**Increase productivity, decrease control**



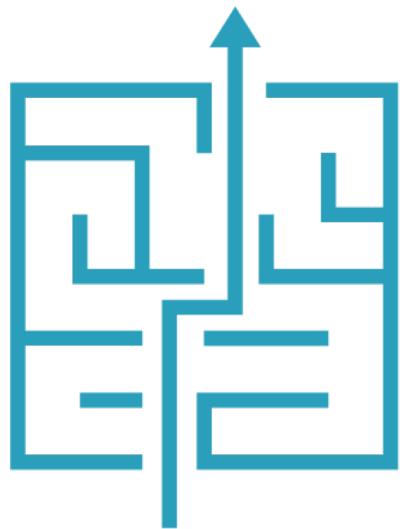
# Form Validation



**Use a form library or implement  
Validate on change or on submission  
Display errors inline or aggregated  
elsewhere**



# Client-side Routing with HTML5 pushState



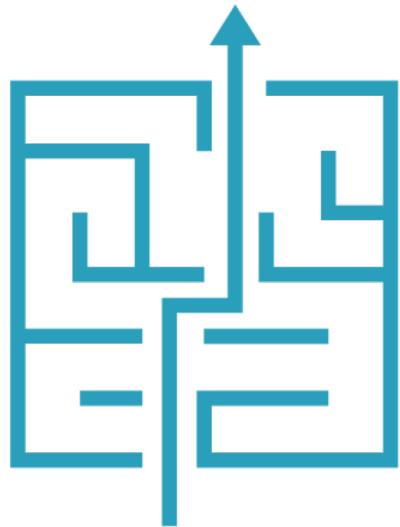
**pushState allows JavaScript to update the browser url**

**Uses proper URLs**

**Requires server-side support**



# React Router



**Client-side router for React**

**Conditional rendering based on routes**

**Supports path updates**



```
<Route exact path="/" component={Welcome} />  
<Route exact path="/about" component={About} />
```

## The Route Component



```
<Route path="/single/:id" component={Single} />
```

## The Route Component



```
function Single({match}) {  
  return <div>{match.params.id}</div>;  
}
```

## The Route Component



# React Router

```
ReactDOM.render(<BrowserRouter>
  <section id="navigation">
    <Route path="/" component={Menu} />
  </section>
  <section id="detail">
    <Route exact path="/" component={Welcome} />
    <Route path="/list" component={List} />
    <Route path="/single/:id" component={Single} />
  </section>
</BrowserRouter>, document.getElementById('app'));
```



# Adding Routes to the Author Quiz

## Author Quiz

Select the book written by the author shown



Macbeth

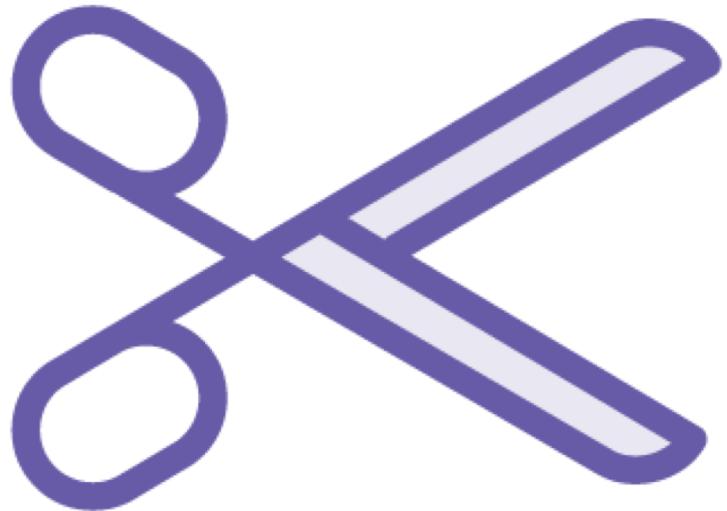
The Shining

Heart of Darkness

Hamlet



# Refs



**Access DOM elements  
Use `React.createRef()`**



# Refs

```
class Identity extends React.Component {  
  constructor() {  
    super();  
    this.myDiv = React.createRef();  
  }  
}
```



# Refs

```
render() {  
  return <div ref={this.myDiv}>  
    {"Set in render <strong>Safe</strong>"}  
  </div>;  
}
```



# Refs

```
componentDidMount() {  
  this.myDiv.current.innerHTML += "<br/> Set on the  
wrapped DOM element. <strong>Unsafe</strong>";  
}
```



# Adding a Form to the Author Quiz

## Author Quiz

Select the book written by the author shown



Macbeth

The Shining

Heart of Darkness

Hamlet



# Summary



**React form support is low level**  
**There are many React form libraries**



## Title or Relevant Graphic

**Animation built in**

**Bullet alternative**

**Room for a bit more text**

**Use this layout for**

- Longer sentence fragments
- List of things
- Procedure list
- Talking points



```
<AuthorQuiz>  
  <Header />  
  <GamePanel data={books} />  
</AuthorQuiz>
```

- ◀ Outer component
- ◀ Child component



# JSX

```
<Sum a={4} b={3} />
```

# JavaScript

```
React.createElement(  
  Sum,  
  {a:4, b:3},  
  null  
)
```



# JSX

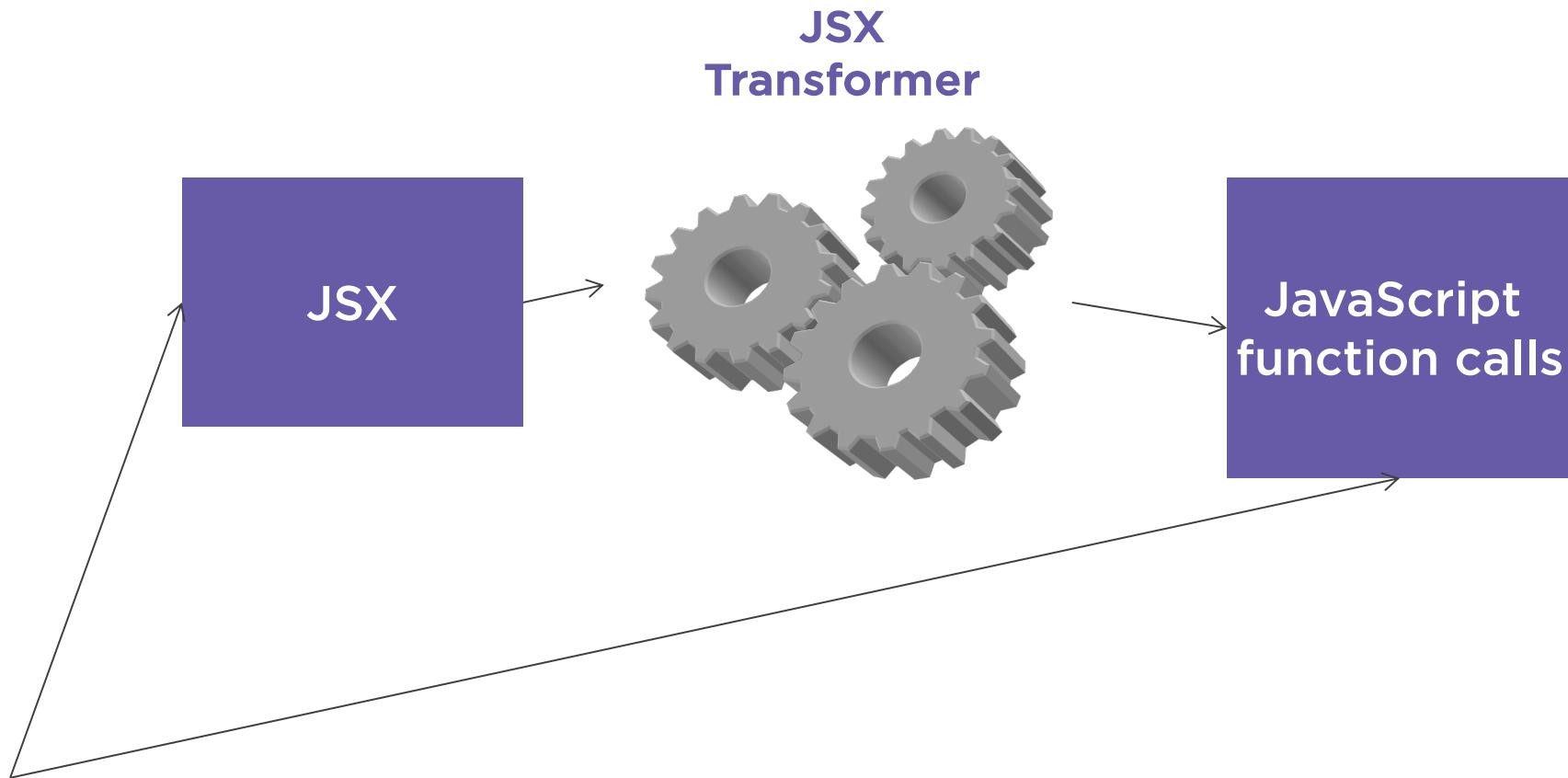
```
<h1>  
  <Sum a={4} b={3} />  
</h1>
```

# JavaScript

```
React.createElement(  
  'h1',  
  null,  
  React.createElement(  
    Sum,  
    {a:4, b:3},  
    null  
  )  
)
```



# Not Using JSX



```
<Hello now={new Date().toISOString()} />
```

Props in JSX  
**JSX attributes become component props**



```
const props = {a: 4, b: 2};  
const element = <Sum {...props} />;
```

## Props in JSX **Spread Attributes**



```
function Clicker({ handleClick }) {  
  return <button onClick={(e)=>{handleClick('A')}}>A</button>  
}  
  
const el = <Clicker handleClick={(l) => {log(l);}} />;
```

## Props in JSX **Events**



```
<label  
  htmlFor="name"  
  className="highlight"  
  style={ { backgroundColor: "yellow" } }>
```

Foo Bar  
</label>

## HTML Attributes

**JSX attributes become HTML attributes**



```
<div dangerouslySetInnerHTML={{__html: "<p>foo</p>" }} />
```

## Unescaping Content

**React escapes content by default**



```
<Hello>  
  <First />  
  <Second />  
</Hello>
```

---

Child Expressions and Elements  
**JSX elements can be nested**



`this.props.children`

---

Child Expressions and Elements

**JSX elements can be nested**



# Child Expressions and Elements

```
function ConditionalDisplay(props) {  
  return <div>  
    {props.isVisible ? props.children : null}  
  </div>;  
}  
  
ConditionalDisplay.propTypes = {  
  isVisible: PropTypes.bool.isRequired  
};
```



# Child Expressions and Elements

```
<ConditionalDisplay isVisible={state.showSum}>  
  <h1>A <span>Sum</span></h1>  
  <Sum a={4} b={2} />  
</ConditionalDisplay>
```



# The Author Quiz

## Author Quiz

Select the book written by the author shown



Macbeth

The Shining

Heart of Darkness

Hamlet



# Summary



**JSX is syntax**

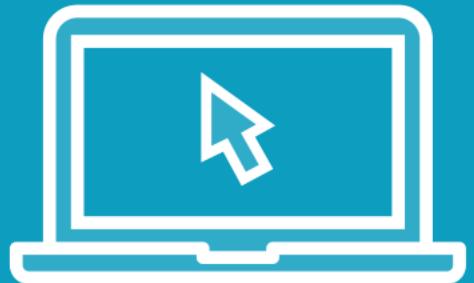
**Component types**

**Props**

**Child components**



Demo



**Setting up a React development environment**

**Building a simple React application**

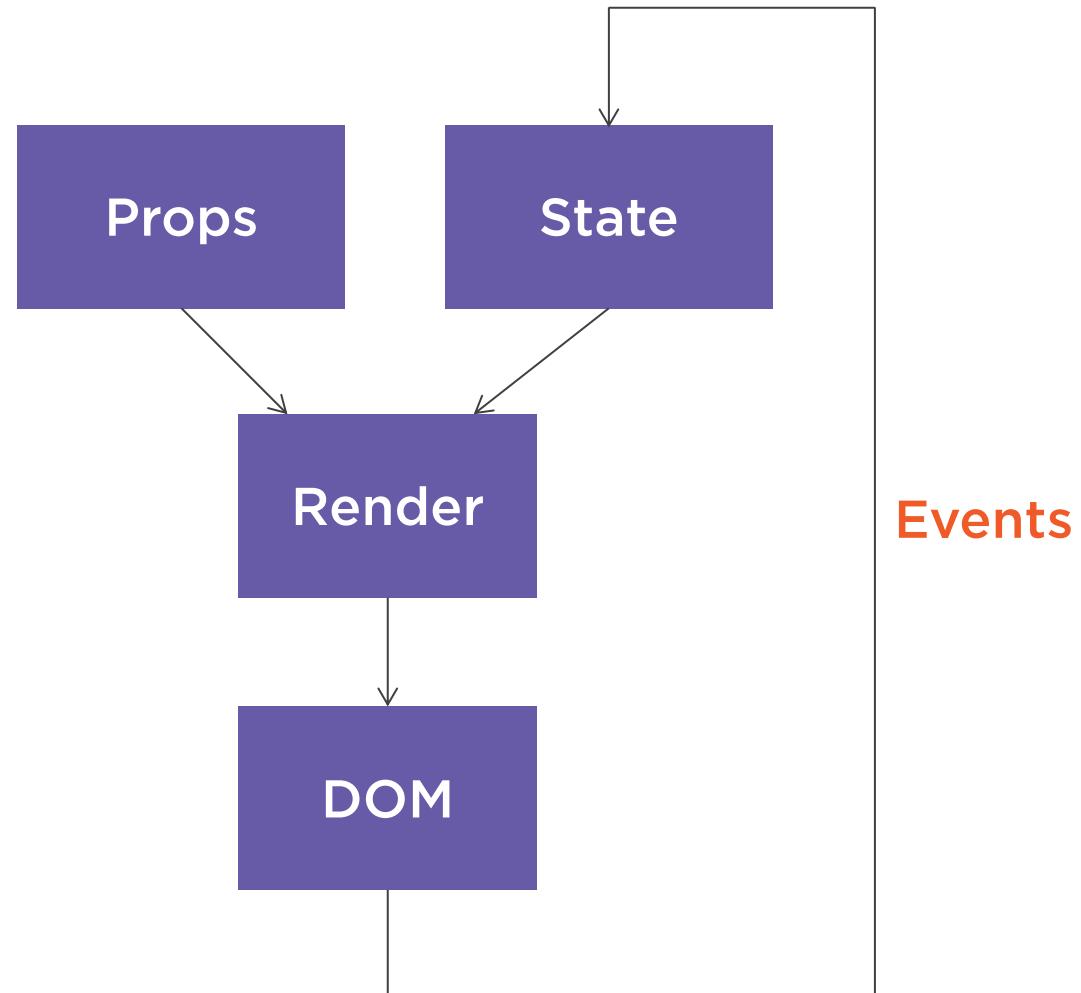


# Advantages and Disadvantages

Advantages	Disadvantages
Conceptual simplicity	Limited in scope
Speed	Productivity
Simple model for server-side rendering	Complex tooling



# Architecture



**Model + Component = DOM**



# React versus Angular

## React

Renders UI and handles events

Uses JavaScript for view logic

JavaScript

## Angular

A complete UI framework

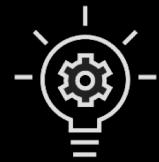
Custom “template expression” syntax

TypeScript









# Using the **Image Chunking Slides**

A presentation slide with a title bar "Click to Add Slide Title in Title Case". Below it are two rows of content. The top row contains two boxes, each with a "Click to add image" placeholder. The bottom row contains two boxes, each with a "Click to add text" placeholder.

Two Image Chunking

A presentation slide with a title bar "Click to Add Slide Title in Title Case". Below it are three rows of content. The top row contains three boxes, each with a "Click to add image" placeholder. The middle row contains three boxes, each with a "Click to add text" placeholder. The bottom row contains three boxes, each with a "Click to add text" placeholder.

Three Image Chunking

A presentation slide with a title bar "Click to Add Slide Title in Title Case". Below it are four rows of content. The top row contains four boxes, each with a "Click to add image" placeholder. The second row contains four boxes, each with a "Click to add text" placeholder. The third row contains four boxes, each with a "Click to add text" placeholder. The bottom row contains four boxes, each with a "Click to add text" placeholder.

Four Image Chunking

A presentation slide with a title bar "Click to Add Slide Title in Title Case". Below it are six rows of content. The first three rows each contain three boxes, with the first two rows having "Click to add image" placeholders and the third row having "Click to add text" placeholders. The last three rows each contain three boxes, all with "Click to add text" placeholders.

Six Image Chunking

These layouts can be used as an alternative to a bulleted list.

They're built specifically for **photos** or **graphics** and look especially awesome when you incorporate icons from the **Pluralsight Icon Library**.

See them in action in the next 4 slides.



# Example of Image Chunking Two Items



**Jill Anderson**

Some information about this graphic goes here and four lines or fewer is best



**John Doe**

Some information about this graphic goes here and four lines or fewer is best



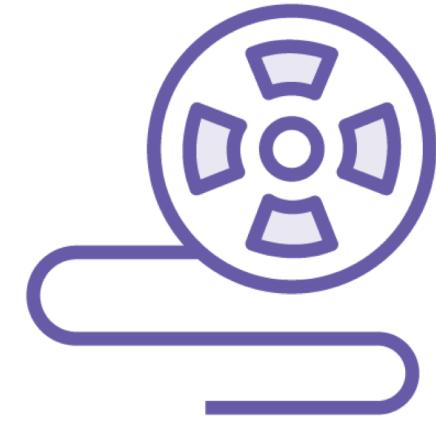
# Example of Image Chunking Three Items



**Clipboard**  
Some information  
goes here; three lines  
or fewer is best



**Book**  
Some information  
goes here; three lines  
or fewer is best



**Film**  
Some information  
goes here; three lines  
or fewer is best



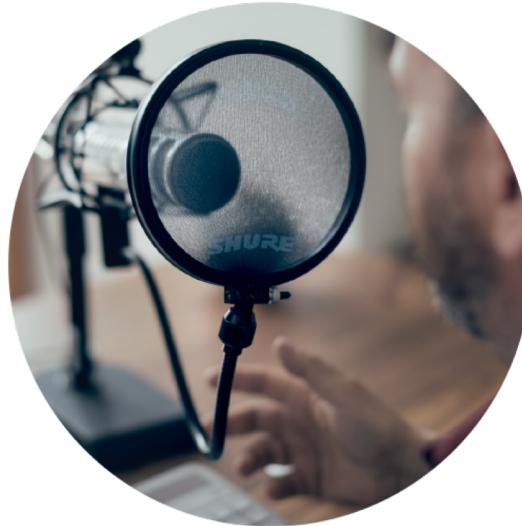
# Example of Image Chunking Four Items



Write



Create



Record



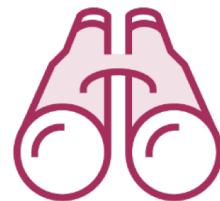
Learn



# Example of Image Chunking Six Items



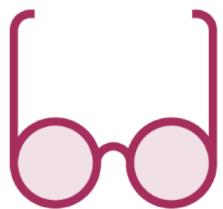
Address book



Binoculars



Camera



Eyeglasses



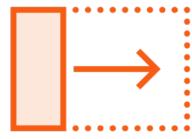
Megaphone



World



# Icons with Bullets



An image chunking option for when you have longer text



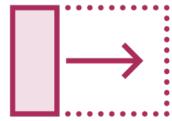
Be concise and keep the text to 3 lines or shorter



Add relative icons on the left for each text item on the right



# Icons with Bullets



An image chunking option for when you have longer text



Two whole lines of text per image also fits nicely however, you should avoid using three lines



Add relative icons on the left for each text item on the right



An alternative to bullets

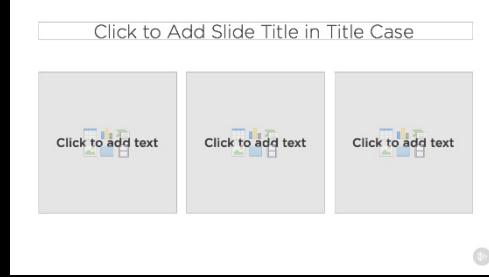




# Using the **Text Chunking Slides**



Two Text Chunking

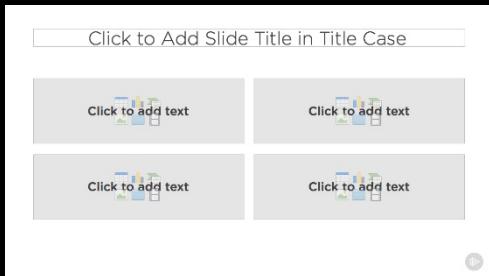


Three Text Chunking

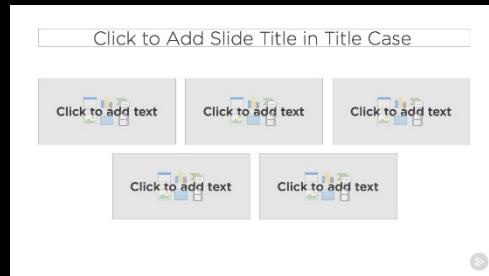
These layouts are intended to group chunks of text. Among other uses, they can be a great alternative to a bullet list.

Use **animations** to bring focus to the point you're speaking on one at a time, and/or use **color** to group points together.

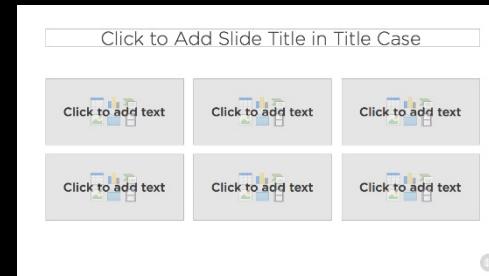
If you have more than six points to discuss, you may want to use a standard bullet list.



Four Text Chunking



Five Text Chunking



Six Text Chunking

We have provided some **example uses** of these layouts in the next few slides.



# Text Chunking Two Items

## Talking point one

Be concise and keep the text  
to four lines or fewer

## Talking point two

Be concise and keep the text  
to four lines or fewer



# Text Chunking Three Items

## Talking point one

Be concise and keep  
the text to four lines  
or fewer

## Talking point two

Be concise and keep  
the text to four lines  
or fewer

## Talking point three

Be concise and keep  
the text to four lines  
or fewer



# Text Chunking Four Items

This is the first talking point  
that should be kept to three  
lines or fewer

This is the second talking  
point that should be kept to  
three lines or fewer

This is the third talking point  
that should be kept to three  
lines or fewer

This is the fourth talking point  
that should be kept to three  
lines or fewer



# Text Chunking Five Items

## Talking point one

Keep the text to three  
lines or fewer

## Talking point two

Keep the text to  
three lines or fewer

## Talking point three

Keep the text to  
three lines or fewer

## Talking point four

Keep the text to  
three lines or fewer

## Talking point five

Keep the text to  
three lines or fewer



# Today's Mobile World

iPhone

Nexus 5

Lumia 950 XL

iPad

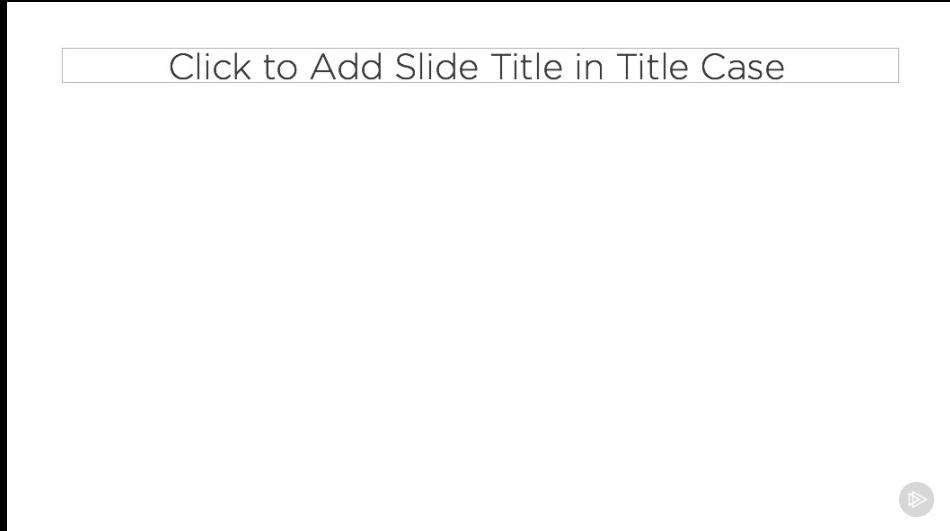
Nexus 7

Surface





# Using the **Title Only Slide**



Title Only

This is the slide you'll want to use when you just need a big space for a diagram, chart, or graphic.

Make sure you check out the training videos available on the **Author Kit** for design best practices.

If you need help bringing your ideas for this space to life, contact your Editor about getting help from one of our **Content Graphic Designers**. In most cases, you just need to submit a rough outline and let our designers work their magic. However, in some special cases, your Editor can get you in touch with a designer directly.

We included some possible starting points for you in the next few slides.



Remember, we are here to help!



# Customer Acquisition and Loyalty

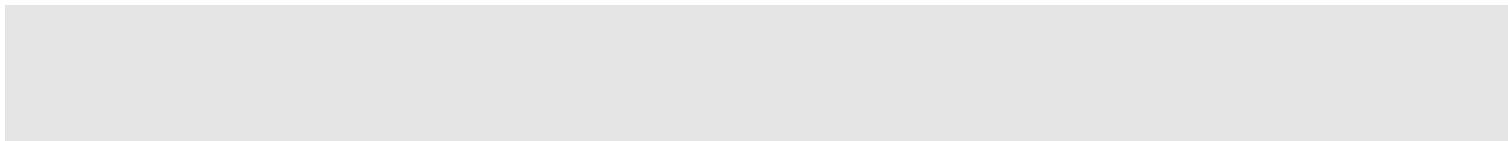
Observed higher sales



42%



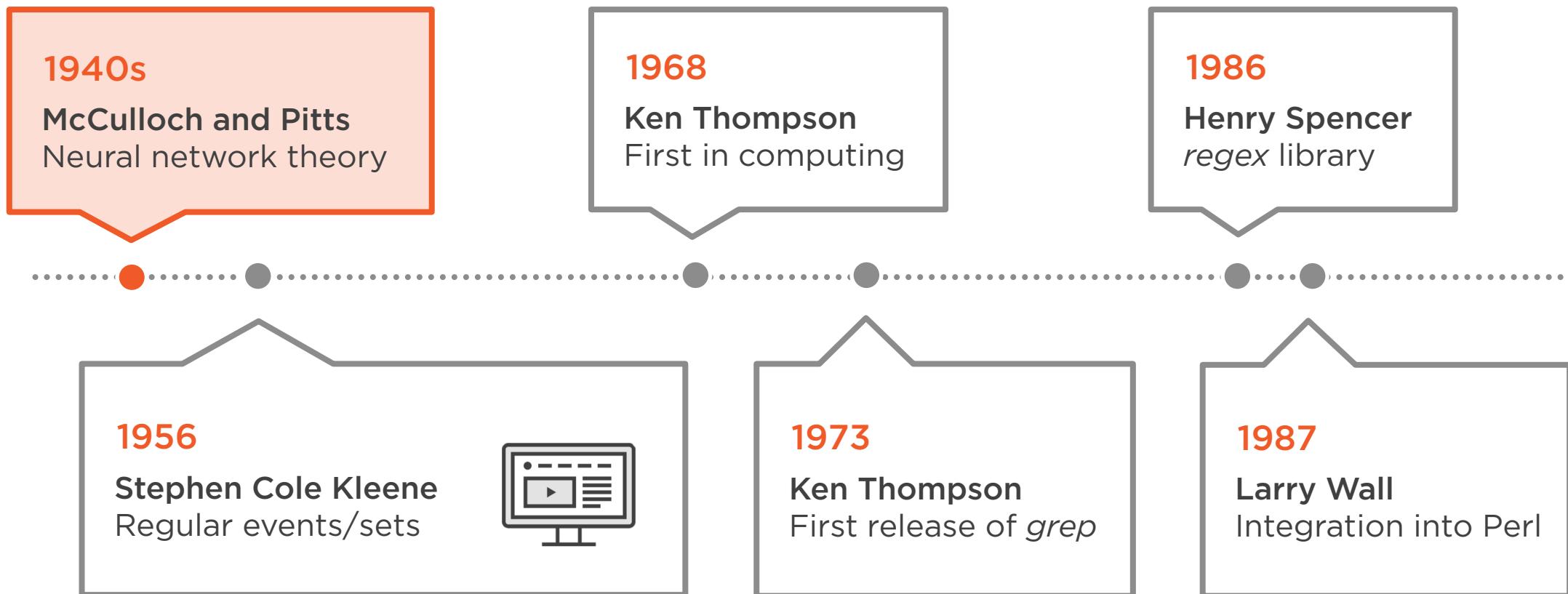
Observed more loyal customers



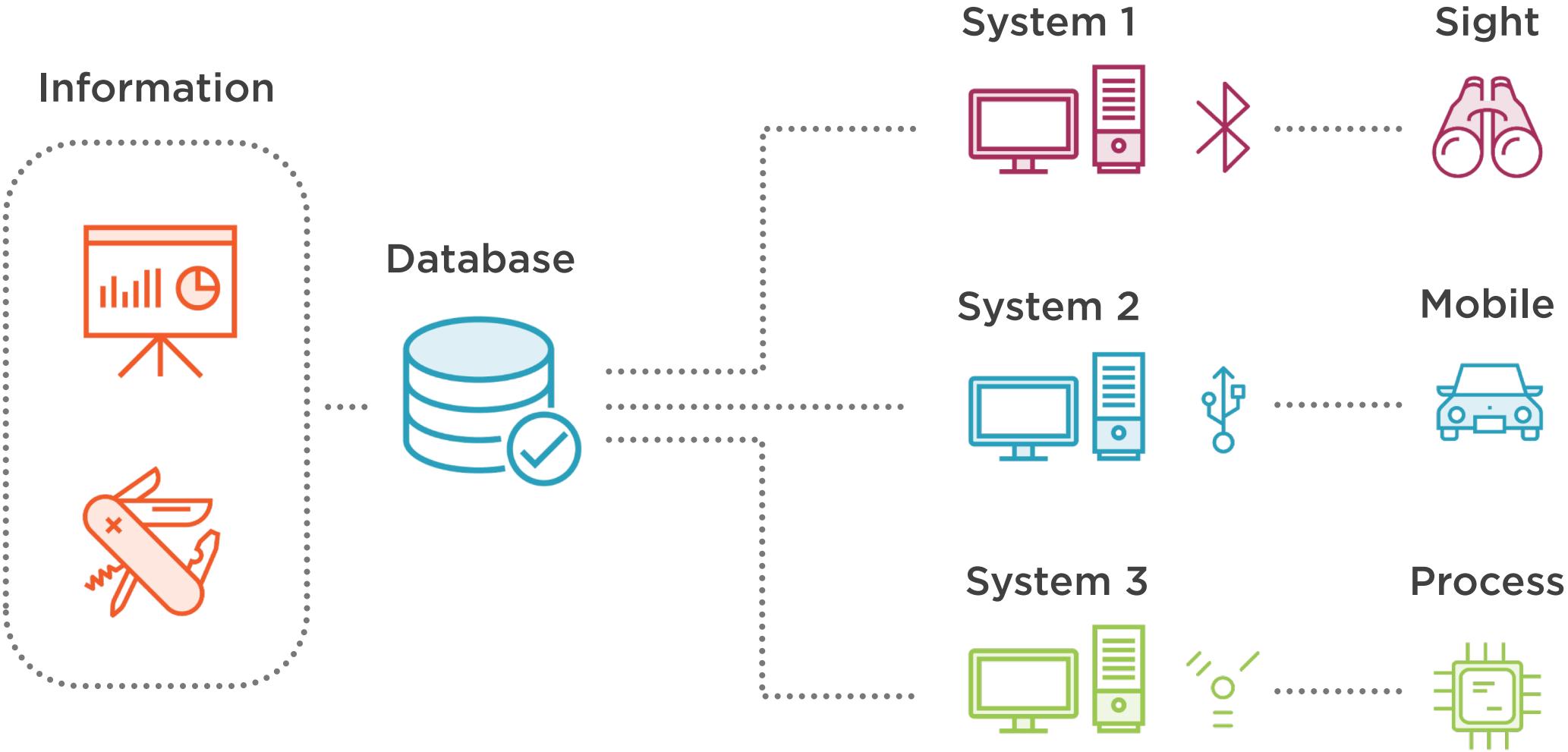
70%



# Timeline of Events



# Title Only Layout Example





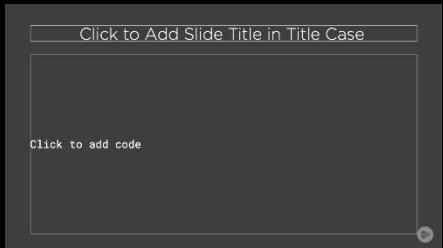
# Using the **Code Slides**



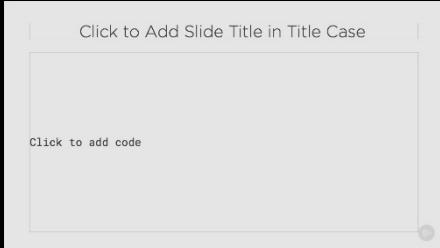
Code Top (Dark)



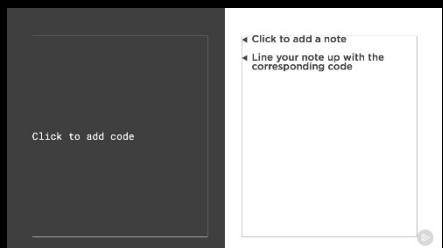
Code Top (Light)



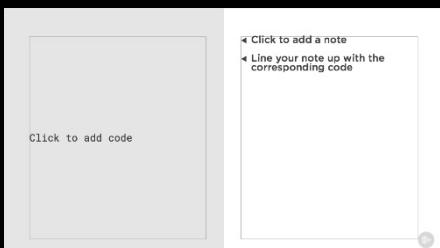
Code (Dark)



Code (Light)



Code Notes (Dark)



Code Notes (Light)

## Code Top Layouts

Use when you need a slide title and info about your code



Make use of the color palette to highlight code.

We recommend using the **Roboto Mono** typeface for your code slides. However, if you use a different font for code in your demos, feel free to use that instead to reinforce a consistent look.

## Code Layouts

Best for larger code snippets

## Code Left Layouts

Great for annotating code structure



```
<div class="row carousel-indicators">  
    <div style="background-color:red;" class="col-md-4" data-target="#homeCarousel" data-slide-to="0" class="active">  
<div class="row carousel-indicators">
```

Slide Title in Titlecase  
**Information about the code above**



```
<div class="row carousel-indicators">  
    <div style="background-color:red;" class="col-md-4" data-target="#homeCarousel" data-slide-to="0" class="active">  
<div class="row carousel-indicators">
```

---

Slide Title in Titlecase  
**Information about the code above**



# Code Snippet on Dark

```
<div class="row carousel-indicators">  
    <div style="background-color:red;" class="col-md-4" data-target="#homeCarousel" data-slide-to="0" class="active">  
        </div>  
    <div style="background-color:green;" class="col-md-4" data-target="#homeCarousel" data-slide-to="1">  
        </div>
```



# Code Snippet on Light

```
<div class="row carousel-indicators">  
    <div style="background-color:red;" class="col-md-4" data-target="#homeCarousel" data-slide-to="0" class="active">  
        </div>  
    <div style="background-color:green;" class="col-md-4" data-target="#homeCarousel" data-slide-to="1">  
        </div>
```



Put code on this side

```
var proto = {  
foo:'Hello World'  
};
```

```
function Bar(){  
Bar.prototype = proto;
```

```
var baz = new Bar();
```

```
console.log(baz.foo);
```

◀ Line up with these notes

◀ Set up prototype object

◀ Constructor function  
and set prototype property

◀ Create instance

◀ Call inherited member



Put code on this side

```
var proto = {  
  foo:'Hello World'  
};
```

```
function Bar(){}  
Bar.prototype = proto;
```

```
var baz = new Bar();
```

```
console.log(baz.foo);
```

◀ Line up with these notes

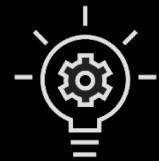
◀ Set up prototype object

◀ Constructor function  
and set prototype property

◀ Create instance

◀ Call inherited member



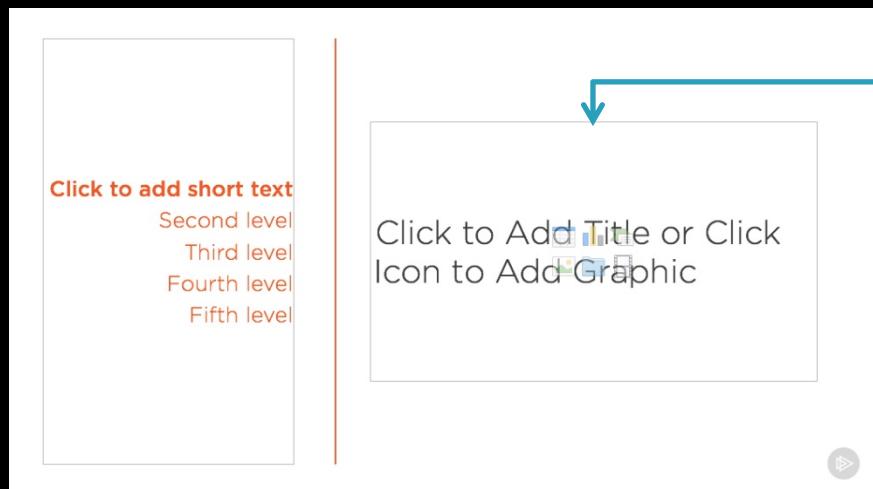


# Using Bullet List Slides

We've provided some bullet list layouts to accommodate various quantities of information.

## Content left | Title/Image right

Intended for bullet text that is shorter and titles/images that are larger

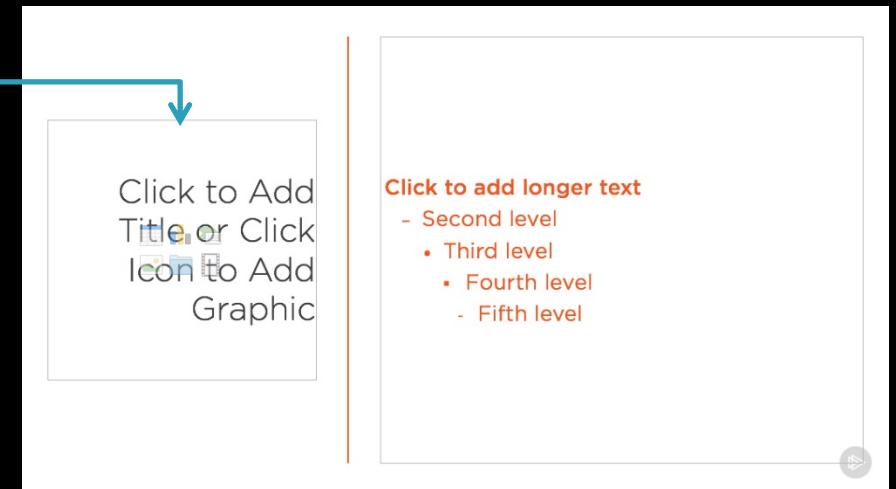


This diagram shows a slide layout with two main sections. On the left, there is a large white box containing placeholder text: "Click to add short text", "Second level", "Third level", "Fourth level", and "Fifth level". To its right is another large white box with the placeholder "Click to Add Title or Click Icon to Add Graphic". A blue arrow points from the text in the first box down to the second box, indicating they are part of the same slide. A small circular navigation icon is at the bottom right.

Content | Image/Title

## Title/Image left | Content right

Intended for bullet text that is longer and titles/images that are smaller



This diagram shows a slide layout with two main sections. On the left, there is a large white box containing placeholder text: "Click to Add Title or Click Icon to Add Graphic", followed by a blue arrow pointing down to another box with the placeholder "Click to add longer text". This second box contains a bulleted list: "- Second level", "• Third level", "• Fourth level", and "- Fifth level". A small circular navigation icon is at the bottom right.

Image/Title | Content



Remember, you can use **text** or **images** in these placeholders.



**Animation built in**

**Bullet alternative**

**Sentence fragments**

**List of things**

**Procedure list**

**Talking points**

Title or Relevant Graphic



**Animation built in**

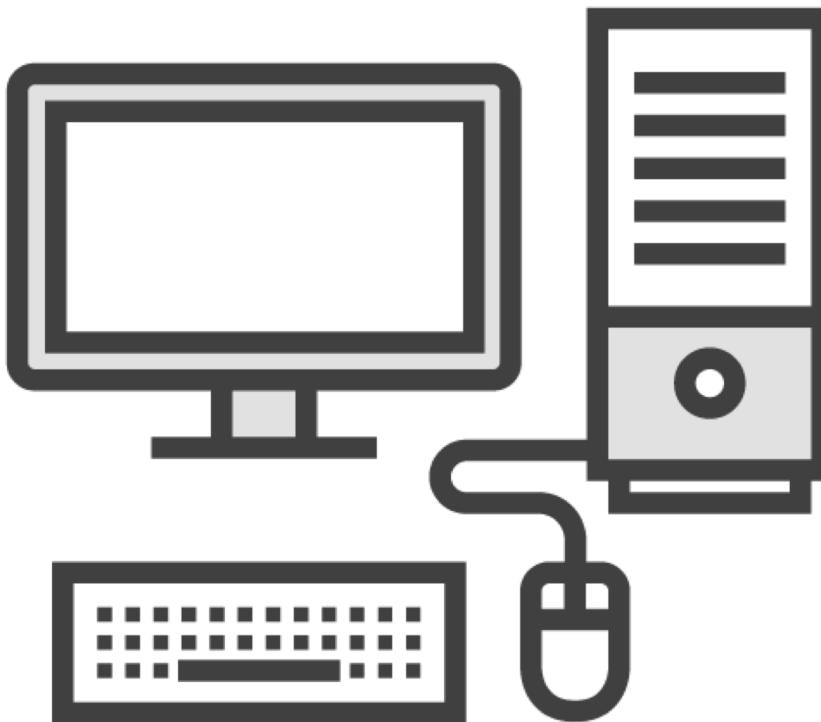
**Bullet alternative**

**Sentence fragments**

**List of things**

**Procedure list**

**Talking points**



## Title or Relevant Graphic

**Animation built in**

**Bullet alternative**

**Room for a bit more text**

**Use this layout for**

- Longer sentence fragments
- List of things
- Procedure list
- Talking points





**Animation built in**

**Bullet alternative**

**Room for a bit more text**

**Use this layout for**

- Longer sentence fragments
- List of things
- Procedure list
- Talking points



# Title Space with Image



**Animation built in**

**Bullet alternative**

**Room for a bit more text**

**Use this layout for**

- Longer sentence fragments
- List of things
- Procedure list
- Talking points





**Graphic on left should fill the entire space**

- Graphic must be high quality and royalty free

**Graphic and text animation is built in**



# Icons with Bullets



Keeping it to one line is best



However, if it is absolutely necessary you can utilize  
the second line if it does not cause a widow



It would look better if you pulled in the box to even it out though



This is a fourth bullet





# Comparison Slide

Use this slide if you need to compare single items or groups of items.

Click to Add Slide Title in Title Case

Compare item one	Compare item two
Click to add text	Click to add text



# Comparison Example

Functional group	Objectives
Configure and administer security	Manage vSphere storage virtualization
Configure advanced networking	Configure software-defined storage
Configure advanced storage	Configure vSphere storage multipathing and failover
Administer and manage resources	Perform advanced VMFS and NFS configurations and upgrades
Configure availability solution	
Deploy and consolidate vSphere	





# Other Slides

The following self-explanatory slides are a good way of adding diversity into the flow of your course.

Use them purposefully.

A white slide template with a large empty rectangular area for content. A thin red horizontal line with the text "Click to Add Section Header in Title Case" is positioned above the main area.

Section Header

A white slide template featuring a green header bar with the text "Word to define". Below it is a smaller white box with the placeholder "Click to add definition".

Definition

A teal slide template with a white rectangular content area containing the text: "This is a short, important statement to bring attention to something."

Important Statement

A white slide template with a pink header bar containing the placeholder "Click to add quote. Don't forget quotation marks". Below it is a smaller white box with the placeholder "Person Quoted".

Quotation



# Section Heading

---



# Word Definition

Here is where you put the definition. This is one of the few places where complete sentences are appropriate. Be sure to cite your source.



This is a short, important statement to bring attention to something.



“Using quotes in your slides can be powerful if used sparingly.”

**Heather Ackmann**

