

Vue Rendering Concepts



Jim Cooper

Software Engineer

@jimthecoop | jcoop.io



What is a Virtual DOM?

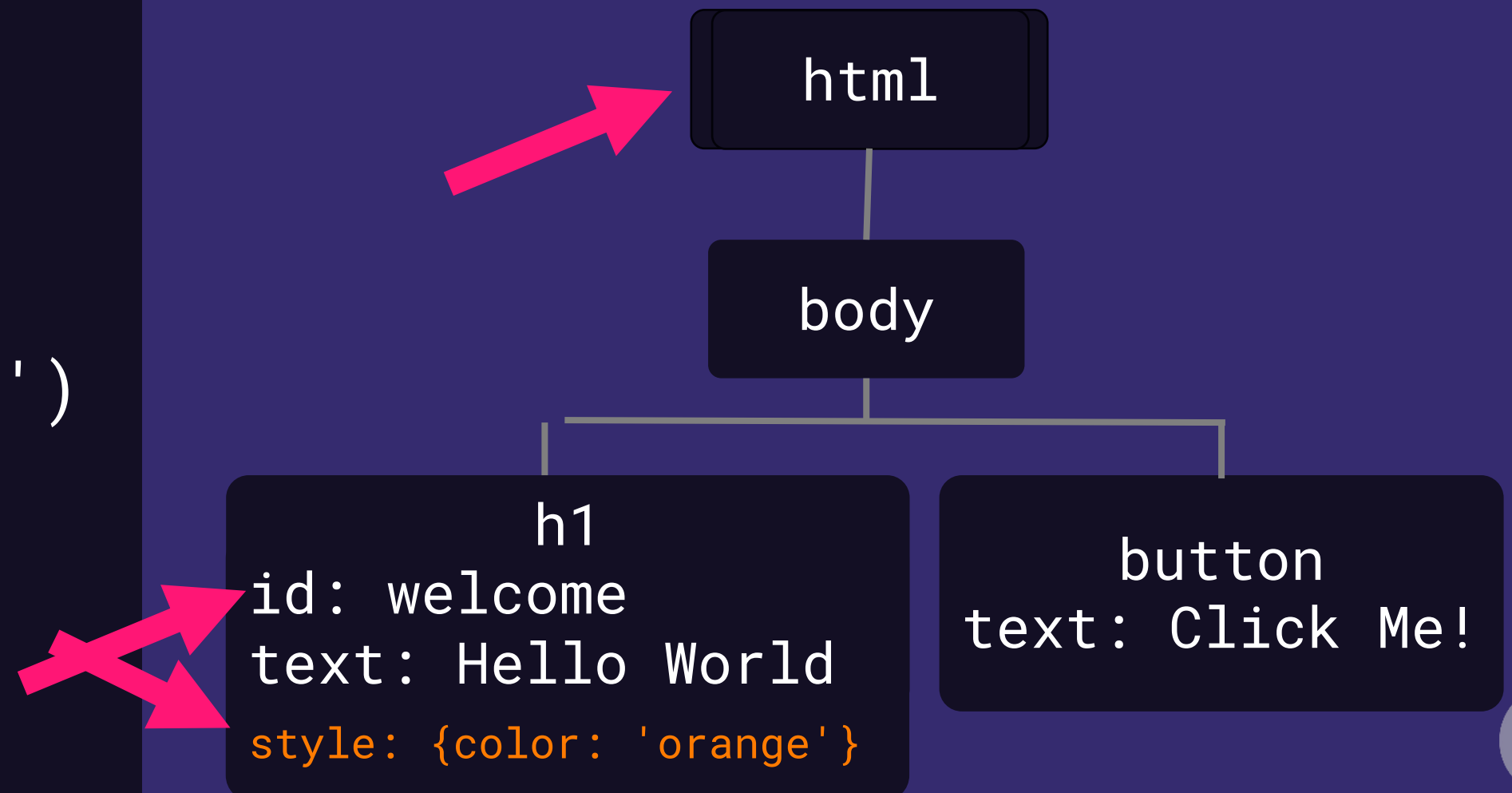
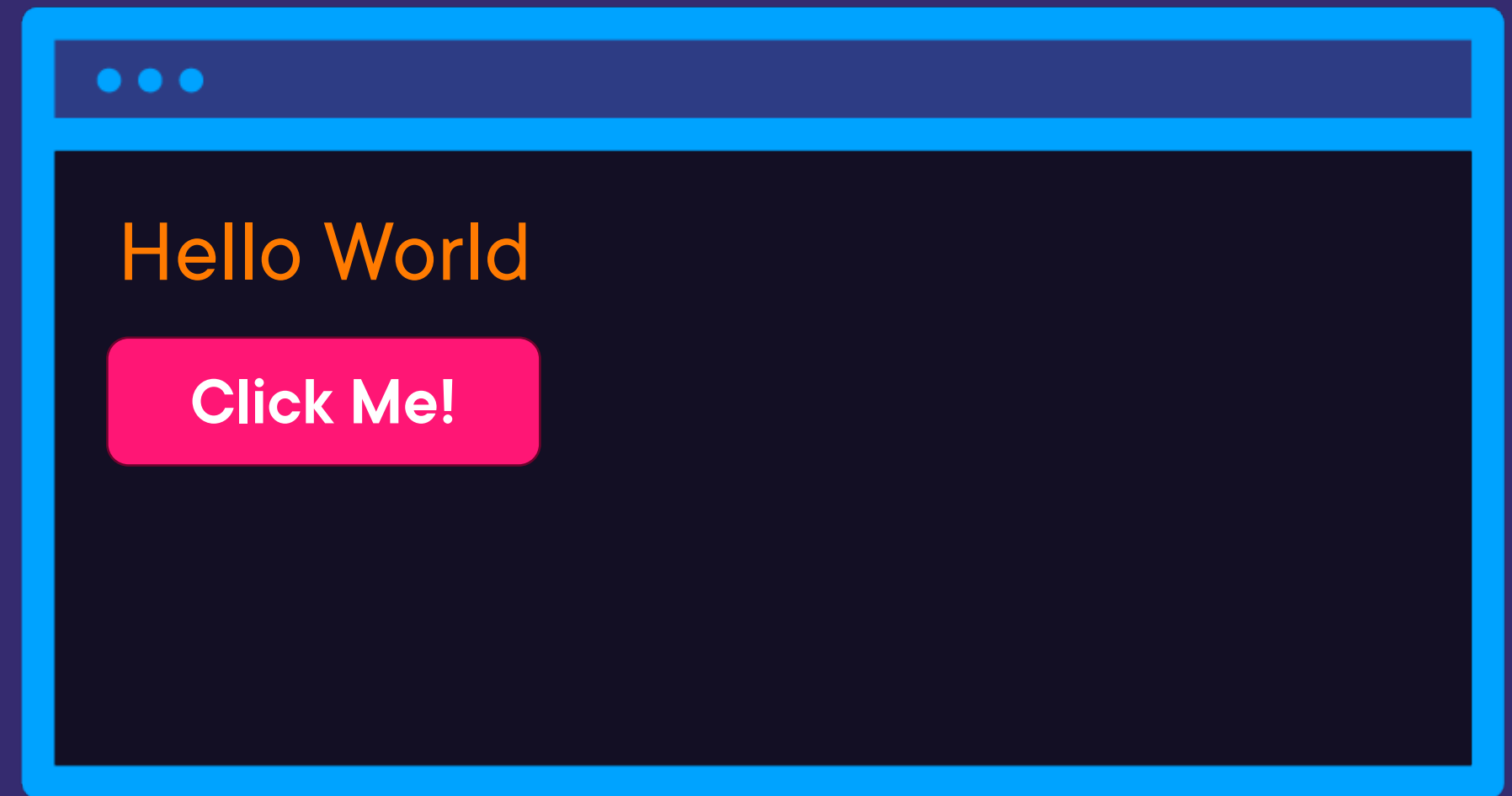


What is the Document Object Model?



```
<html>
  <body>
    <h1 id="welcome">
      Hello World
    </h1>
    <button>Click Me!</button>
  </body>
</html>
```

```
document.getElementById('welcome')
  .style.color = 'orange';
```

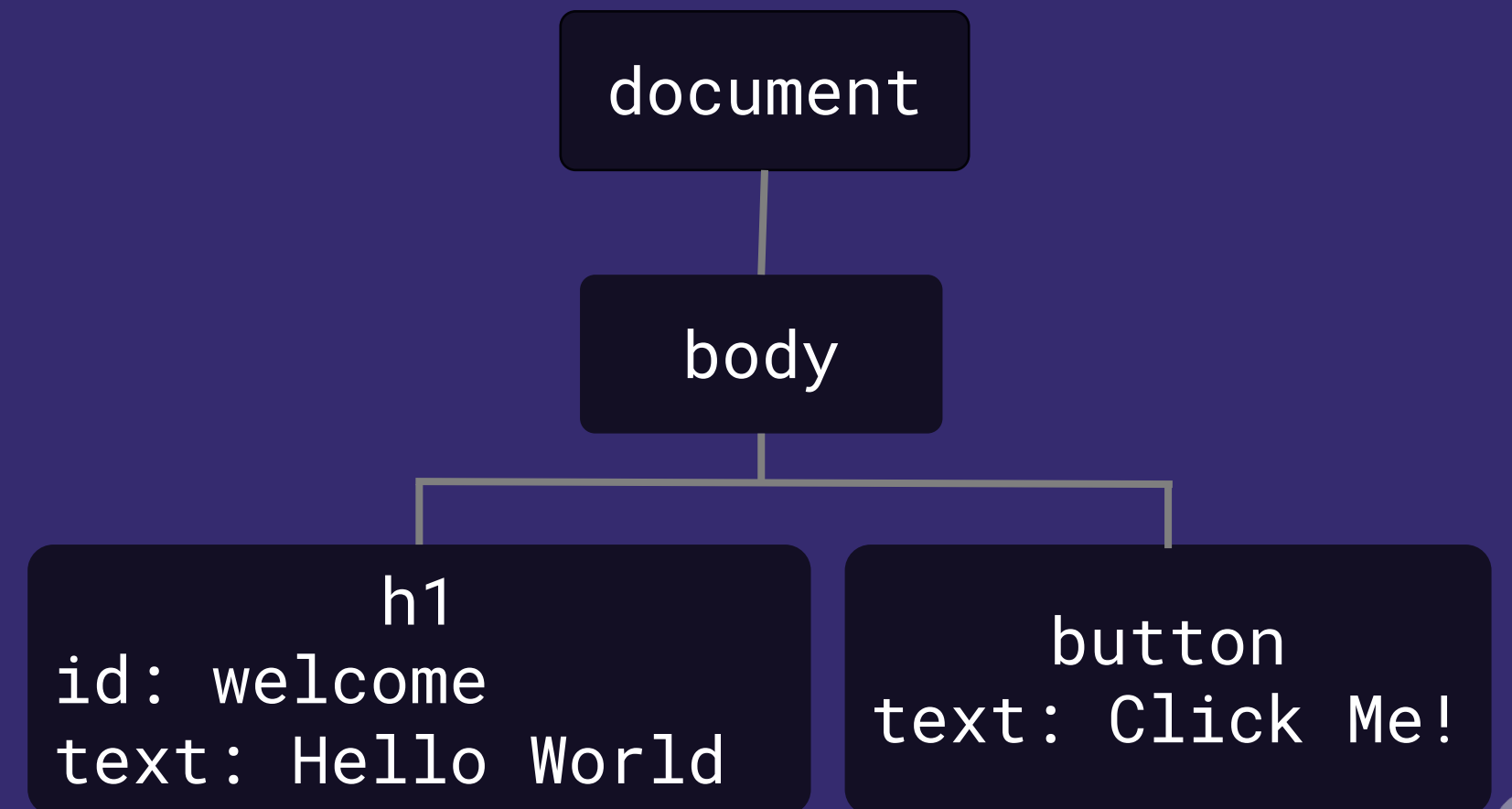
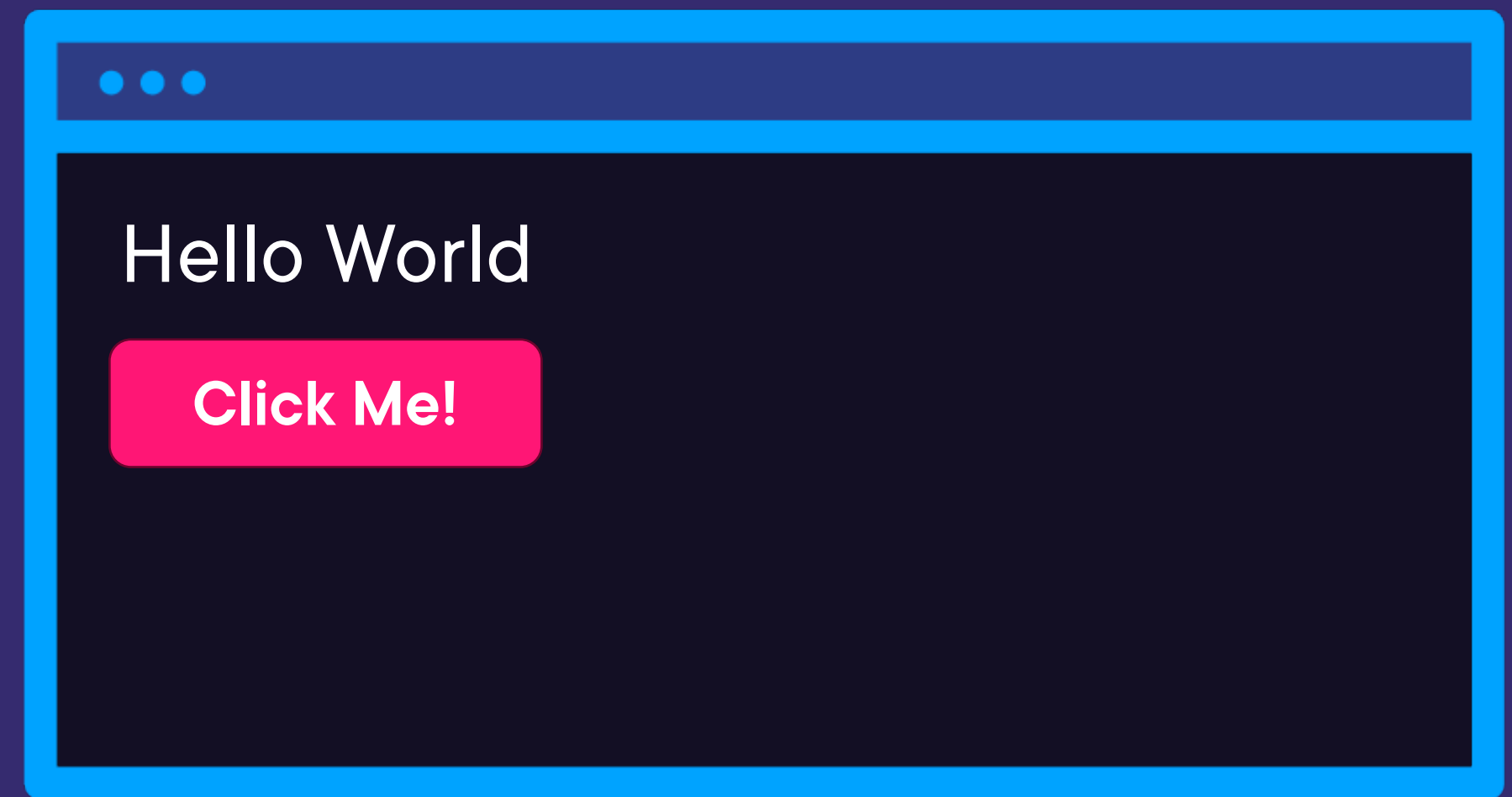


What is a Virtual DOM?

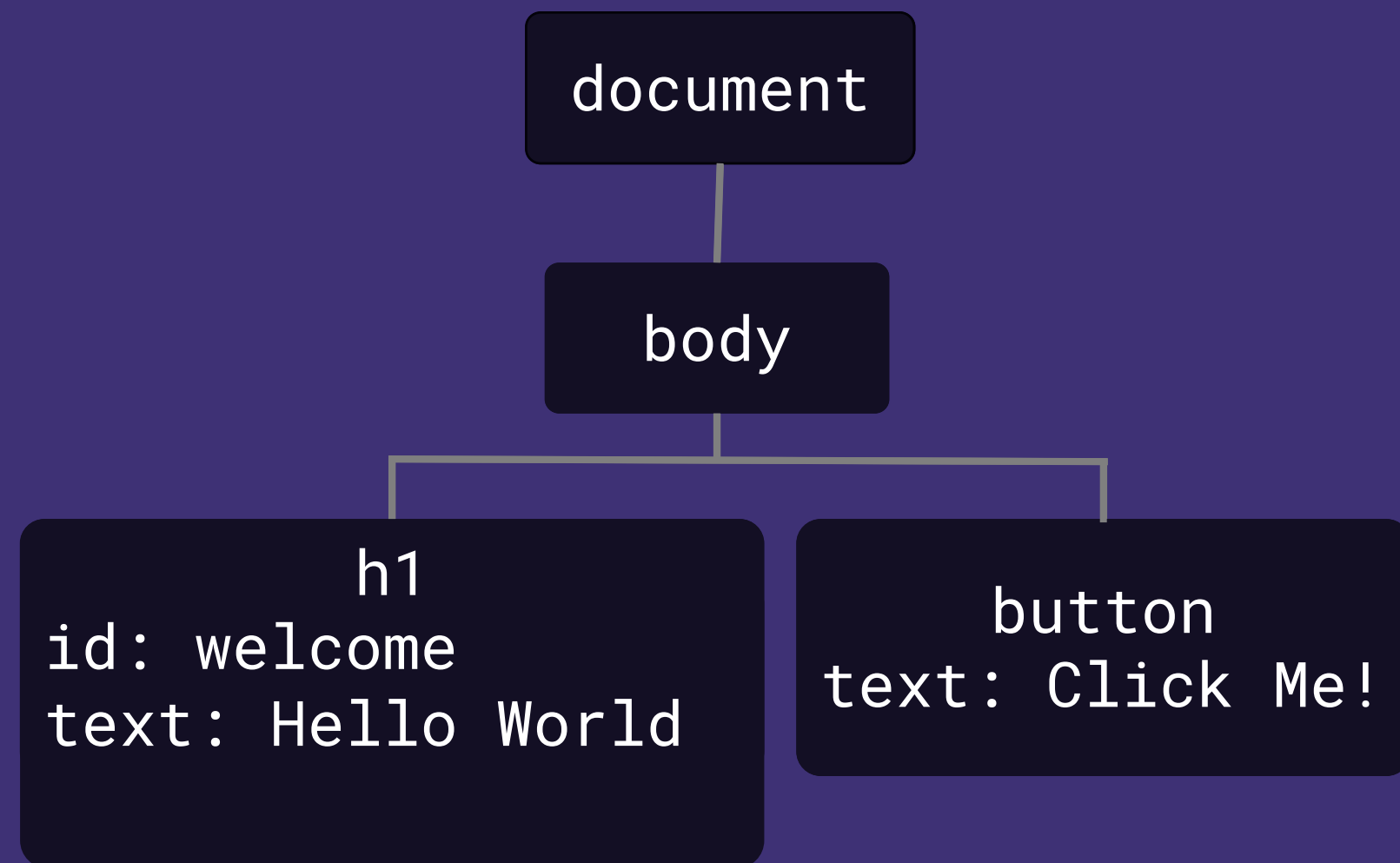


```
<html>
  <body>
    <h1 id="welcome">
      Hello World
    </h1>
    <button>Click Me!</button>
  </body>
</html>
```

```
document.getElementById('welcome')
  .style.color = 'orange';
```



DOM



Virtual DOM

```
style: {color: 'orange'}
```



The Render Pipeline

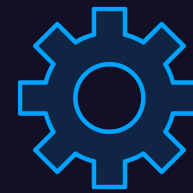
MyComponent.vue

```
<template>
  <h1 class="header">Hello</h1>
  <div>{{name}}</div>
  <button @click="handleClick()">
    Click Me!
  </button>
</template>
```

```
<script setup>
  const userName = ref('Jim')
  const handleClick = () => { }
</script>
```

```
<style scoped>
.header { font-size: 18px; }
</style>
```

Mount



Compiled

```
import { h } from 'vue'

function render() {
  const vnode1 = h('h1',
    { class: 'header' },
    'Hello')

  const vnode2 = h('div',
    { },
    userName)

  const vnode3 = h('button',
    { onClick: 'handleClick()' },
    'Click Me!')

  return [vnode1, vnode2, vnode3]
}
```



The Render Pipeline

Render Function

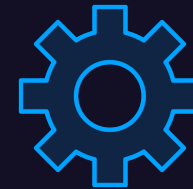
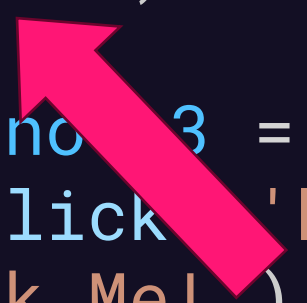
```
import { h } from 'vue'

function render() {
  const vnode1 = h('h1',
    { class: 'header' },
    'Hello')

  const vnode2 = h('div',
    { },
    userName)

  const vnode3 = h('button',
    { onClick: 'handleClick()' },
    'Click Me!')

  return [vnode1, vnode2, vnode3]
}
```



Generates

Virtual DOM

```
const vDOM = [
  {
    type: 'h1',
    props: { class: 'header' },
    children: ['Hello']
  },
  {
    type: 'div',
    props: {},
    children: ['Jim']
  },
  {
    type: 'button',
    props: {
      onClick: handleClick()
    },
    children: ['Click Me!']
  }
]
```



The Render Pipeline

Render Function

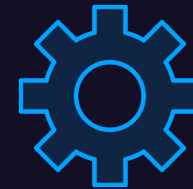
```
import { h } from 'vue'

function render() {
  const vnode1 = h('h1',
    { class: 'header' },
    'Hello')

  const vnode2 = h('div',
    { },
    userName)

  const vnode3 = h('button',
    { onClick: 'handleClick()' },
    'Click Me!')

  return [vnode1, vnode2, vnode3]
}
```



Generates

Virtual DOM 2

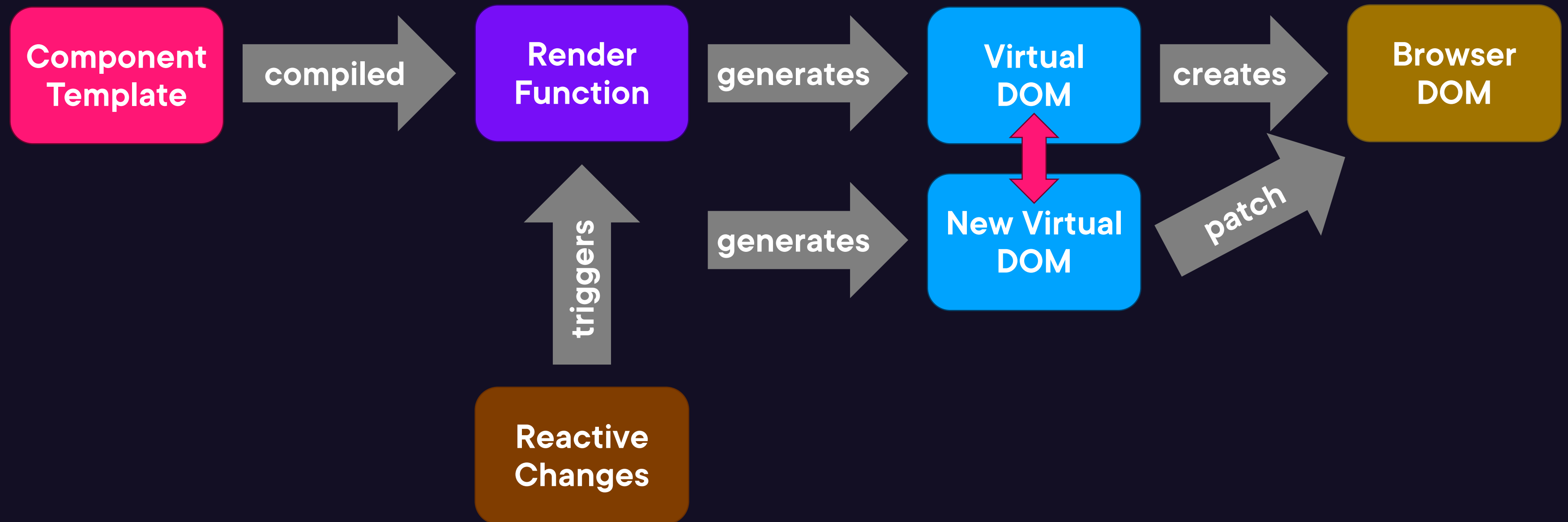
```
const vDOM = [
  {
    type: 'h1',
    props: { class: 'header' },
    children: ['Hello']
  },
  {
    type: 'div',
    props: {},
    children: ['Kris']
  },
  {
    type: 'button',
    props: {
      onClick: handleClick()
    },
    children: ['Click Me!']
  }
]
```

Virtual DOM 1

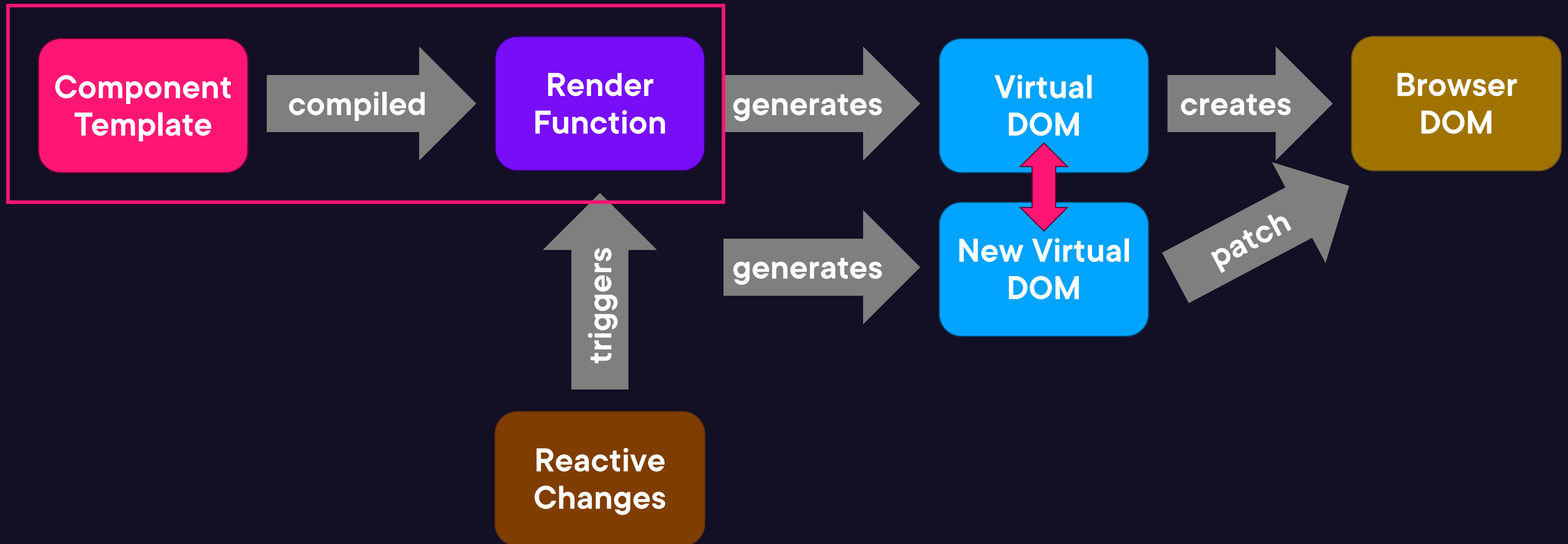
```
const vDOM = [
  {
    type: 'h1',
    props: { class: 'header' },
    children: ['Hello']
  },
  {
    type: 'div',
    props: {},
    children: ['Jim']
  },
  {
    type: 'button',
    props: {
      onClick: handleClick()
    },
    children: ['Click Me!']
  }
]
```



The Render Pipeline



The Render Pipeline



Using Render Functions

TraditionalComponent.vue

```
<template>
  <h1 class="header">Hello</h1>
  <div>{{name}}</div>
  <button @click="handleClick()">
    Click Me!
  </button>
</template>
```

```
<script setup>
  const userName = ref('Jim')
  const handleClick = () => { }
</script>
```

RenderComponent.vue

```
<script>
import { h } from 'vue'

function render() {
  const userName = ref('Jim')
  const handleClick = () => { }

  const vnode1 = h('h1',
    { class: 'header' },
    'Hello')

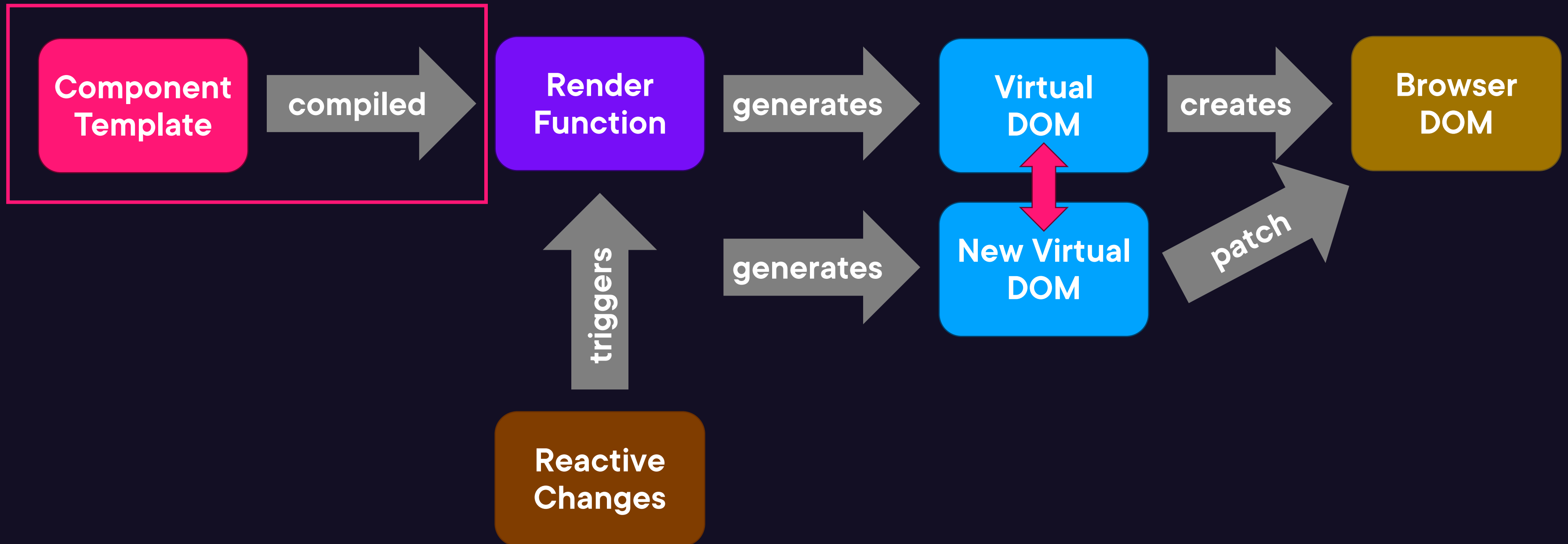
  const vnode2 = h('div',
    { },
    userName)

  const vnode3 = h('button',
    { onClick: 'handleClick()' },
    'Click Me!')

  return [vnode1, vnode2, vnode3]
}
```



The Render Pipeline



Fallthrough Attributes

```
<MyComponent
```

```
  class="red"
```

```
  :product="myProduct"
```

```
  @click="handleClick()"
```

```
/>
```

```
<ChildComponent />
```

```
<ChildComponent2 />
```



Fallthrough Attributes

```
<MyComponent
```

```
/>
```

```
<ChildComponent />
```

```
class="red"
```

```
:product="myProduct"
```

```
@click="handleClick()"
```

```
<ChildComponent2 />
```



Fallthrough Attributes

```
<MyComponent
```

```
/>
```

```
<ChildComponent />
```

```
class="red"
```

```
:product="myProduct"
```

```
<ChildComponent2 />
```

```
@click="handleClick()"
```

