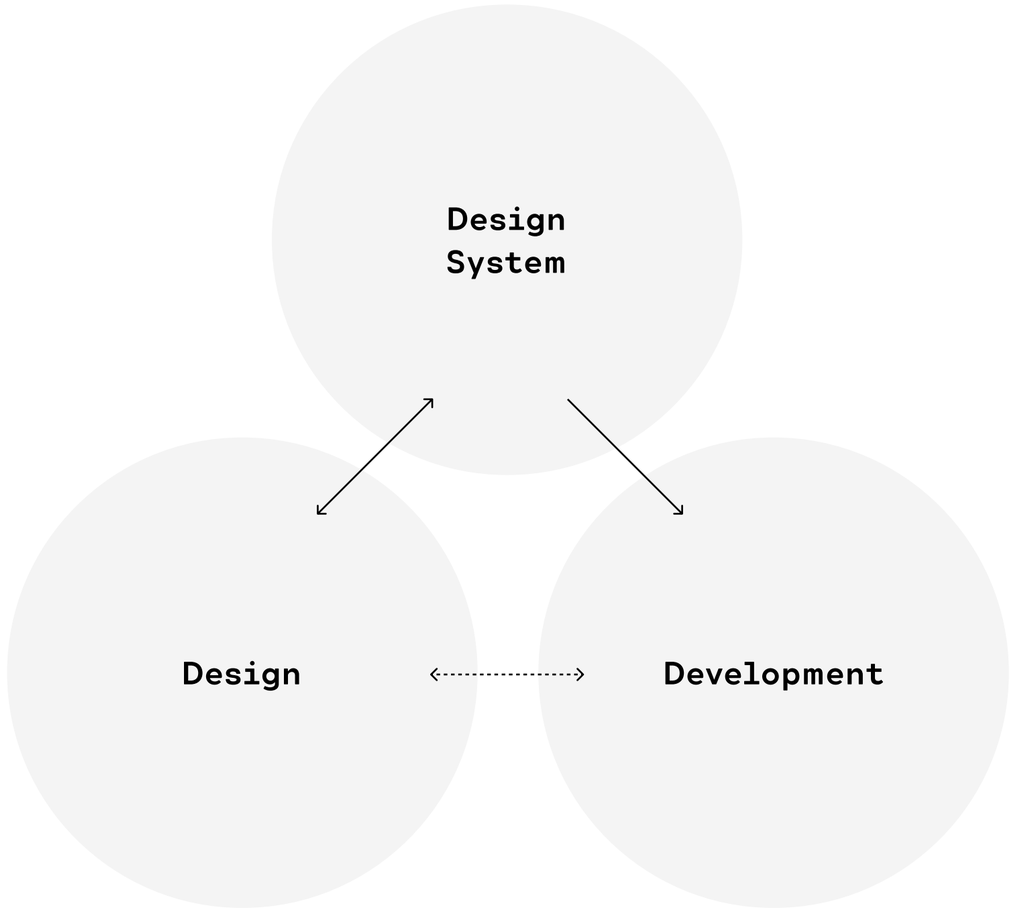
Patrick's Design System Philosophy

v0.1



Design

Who: Designers

Use Figma/Sketch/other with a component library that maps 1:1 to the design system's tokens, atoms, components, and layouts.

Any new design system concepts are explored in design. Any new concepts must earn their way into the design system.

Design Tokens

Colors

Box Shadow

Font Size

Medium Font Size

Atoms

Heading Level 1

This is a headline

Components

Card

Card Headline

This is card content.

Layouts

Grid

Design System

Who: Development-savvy designers and/or design-savvy developers.

Implement designers' mockups into a resuable, scalable design system. This is the source of truth for all design usage and implementation with HTML and CSS in production-level code.

The design system is consistently maintained and documented for developer and designer reference.

Design Tokens

```
--blue-500: #3255AD;
--box-shadow-medium: 0 2px 2px rgba(0, 0, 0, 0.1);
--font-medium: 1rem;
--spacing-2: 0.5rem;
--spacing-4: 1rem;
```

Atoms

HTML

```
<h1 class="h1">Heading 1</h1>
```

CSS

```
.h1 {
  font-size: var(--font-xlarge);
  line-height: 1.25;
  font-weight: bold;
  margin: 0;
}
```

Components

HTML

```
<div class="card">
  <div class="card__header">
    <h1 class="h1">Card Headline</h1>
  </div>
  <div class="card__content">
    <p>This is card content.</p>
  </div>
</div>
```

CSS

```
.card {
  background: var(--white);
  box-shadow: var(--box-shadow-small);

  &__header {
    padding: var(--spacing-4);
    border-bottom: 1px solid var(--gray-300);
  }

  &__content {
    padding: var(--spacing-4);
  }
}
```

Layouts

HTML

```
<div class="grid grid--3-col">
  <Card />
  <Card />
  <Card />
</div>
```

CSS

```
.grid {
  display: grid;
  gap: var(--spacing-2);

  &--3-col {
    grid-template-columns: repeat(3, 1fr);
  }
}
```

Utility classes

```
.u-text-bold {
  font-weight: bold !important;
}

.u-text-red-500 {
  color: var(--red-500); !important;
}
```

Development

Who: Developers

Reference the Design System documentation to build consistent user interfaces. Does not author CSS. None. Zero. Zilch.

Development is in constant communication with Design System and Design team(s) to verify and get sign off on usage.

Design Tokens

Don't even need to think about it.

Atoms

HTML

```
<h1 class="h1">{{ page.title }}</h1>
```

Components

HTML

```
<div class="card">
  <div class="card__header">
    <h1 class="h1">{{ card.title }}</h1>
  </div>
  <div class="card__content">
    <p>{{ card.content }}</p>
  </div>
</div>
```

Layouts

HTML

```
<div class="grid grid--3-col">
  <Card />
  <Card />
  <Card />
</div>
```

Utility classes

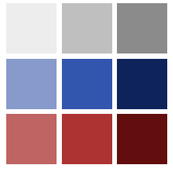
Use for one-off adjustments that dont' require changes to the whole design system.

Fully communicate to design system team that these are being used. If a pattern emerges from repeat, consistent usage, it can be incorporated properly into the design system.

Design Tokens

CSS custom properties and/or Sass variables. Used to set consistent values throughout a product.

Colors



```
--red-100: #C06363;
--red-500: #AD3232;
--red-900: #620E0E;
```

Spacing

```
--spacing-1: 0.25rem;
--spacing-2: 0.5rem;
--spacing-3: 0.75rem;
--spacing-4: 1rem;
--spacing-6: 1.5rem;
--spacing-8: 2rem;
```

Font sizes

Smallest

Small

Medium Font Size

Large

Extra Large

```
--font-smallest: 0.75rem;
--font-small: 0.875rem;
--font-medium: 1rem;
--font-large: 1.125rem;
--font-xlarge: 1.5rem;
```

Other Design Tokens

- Box shadows
- Animation easings
- Animation timings
- et. al.

Atoms

Small, single block items with potentioal modifiers. Generally used within larger componenets, but can be used on their own.

Typography

Heading Level 1

This is a headline

Heading Level 2

This is a secondary headline

HTML

```
<h1 class="h1">This is a headline.</h1>
<h2 class="h2">This is a secondary headline</h2>
```

CSS

```
.h1 {
  font-size: var(--font-xlarge);
  line-height: 1.25;
  font-weight: bold;
  margin: 0;
}

.h2 {
  font-size: var(--font-large);
  line-height: 1.25;
  margin: 0;
}
```

Icons

HTML

```
<svg class="icon">...</svg>
```

CSS

```
.icon {
  display: inline-block;
  width: 1em;
  height: 1em;
  fill: var(--gray-500);
}
```

Form Elements

Form Label

Full Name

Text Input

Patrick Hildebrandt

HTML

```
<label class="form-label">Full Name</label>
<input type="text" class="form-input" name="full_name" />
```

CSS

```
.form-label {
  font-weight: bold;
  color: var(--gray-600);
}

.form-input {
  font-size: var(--font-medium);
  border: 1px solid var(--gray-200);
  padding: var(--spacing-2) var(--spacing-4);
}
```

Components

More complex than atoms. Fully-composed BEM. Can house atoms or other components.

Buttons

Simple Button

Button

Button with icon

+ Button

HTML

```
<button class="button">Button</button>

<button class="button">
  <svg class="button__icon icon">...</svg>
  Button
</button>
```

CSS

```
.button {
  background: var(--blue-500);
  color: var(--white);
  font-weight: bold;
  padding: var(--spacing-2) var(--spacing-4);

  &__icon {
    margin-right: 0.25rem;
  }
}
```

Cards

Card Headline

This is card content.

+ Button

HTML

```
<div class="card">
  <div class="card__header">
    <h1 class="h1">Card Headline</h1>
  </div>
  <div class="card__content">
    <p>This is card content.</p>
  </div>
  <div class="card__actions">
    <button class="button">
      <svg class="button__icon icon">...</svg>
        Button
    </button>
  </div>
</div>
```

CSS

```
.card {
  background: var(--white);
  box-shadow: var(--box-shadow-small);

  &__header {
    padding: var(--spacing-4);
    border-bottom: 1px solid var(--gray-300);
  }

  &__content {
    padding: var(--spacing-4);
  }
}
```

Layouts

Used to layout components on a page. Layouts can be sub-sections of the page or the overall structure of the page.

Reusable Grids

HTML

```
<div class="grid grid--3-col">
  <Card />
  <Card />
  <Card />
</div>
```

CSS

```
.grid {
  display: grid;
  gap: var(--spacing-2);

  &--3-col {
    grid-template-columns: repeat(3, 1fr);
  }
}
```

Page Layouts

HTML

```
<div class="page">
  <header class="page__header">
    <PageHeader />
  </header>
  <div class="page__sidebar">
    <PageSidebar />
  </div>
  <div class="page__content">
    <PageContent />
  </div>
</div>
```

CSS

```
.page {
  display: grid;
  grid-template-columns: 300px 1fr;
  grid-template-areas:
    "header header"
    "sidebar content";

  &__header {
    grid-area: header;
  }

  &__sidebar  {
    grid-area: sidebar;
  }

  &__content {
    grid-area: content;
  }
}
```

Utility Classes

Used for one-off nudging, adjusting, and overriding when the design system doesn't accommodate. Utility classes should be used sparingly.

Margin & Padding

```
.u-ma-2 {
  margin: 0.5rem !important;
}

.u-mb-4 {
  margin-bottom: 1rem !important;
}

.u-px-8 {
  padding-left: 2rem !important;
  padding-right: 2rem !important;
}
```

Text

```
.u-text-bold {
  font-weight: bold !important;
}

.u-text-red-500 {
  color: var(--red-500); !important;
}
```

UTILITY CLASS WARNING

!important is used in utility classes as a means to spike their specificity since they can be used to override component-level properties.

Utility classes are not a means to construct new components. Also, if a pattern emerges where utility classes are being consistently applied to components that new pattern should be incorporated into the design system as a component modifier.