



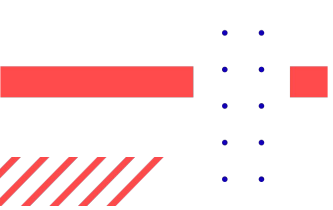
Web开发进阶班第1期

Lesson 4 CSS & SASS

2023.07.25

www.jiangren.com.au

1. CSS Flexbox
2. Responsive Web Design
3. SASS (Css Preprocessor)
4. Design Principles



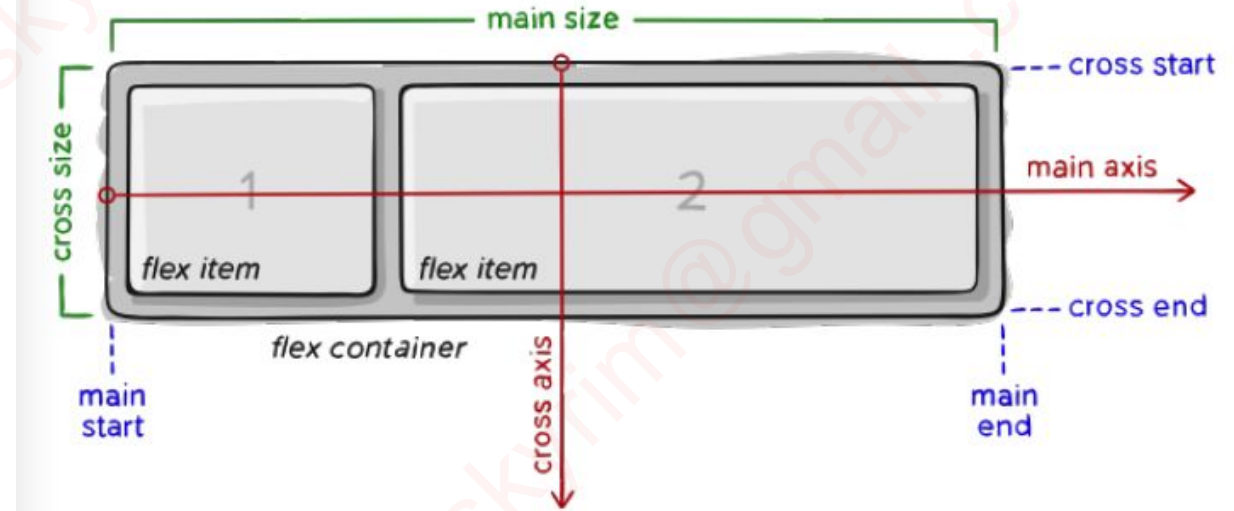
1. Flexbox

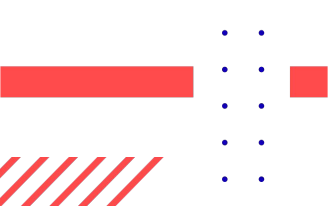
- The Flexbox Layout (Flexible Box) module (a W3C Candidate Recommendation as of October 2017) aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word “flex”).
- The main idea behind the flex layout is to give the container the ability to alter its items’ width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes). A flex container expands items to fill available free space or shrinks them to prevent overflow.
- A flex container expands items to fill available free space, or shrinks them to prevent overflow.

Demo: <https://codepen.io/enxaneta/full/adLPwv>

1. Flexbox

- **main axis** – The main axis of a flex container is the primary axis along which flex items are laid out. Beware, it is not necessarily horizontal; it depends on the `flex-direction` property (see below).
- **main-start | main-end** – The flex items are placed within the container starting from main-start and going to main-end.
- **main size** – A flex item's width or height, whichever is in the main dimension, is the item's main size. The flex item's main size property is either the 'width' or 'height' property, whichever is in the main dimension.
- **cross axis** – The axis perpendicular to the main axis is called the cross axis. Its direction depends on the main axis direction.
- **cross-start | cross-end** – Flex lines are filled with items and placed into the container starting on the cross-start side of the flex container and going toward the cross-end side.
- **cross size** – The width or height of a flex item, whichever is in the cross dimension, is the item's cross size. The cross size property is whichever of 'width' or 'height' that is in the cross dimension.





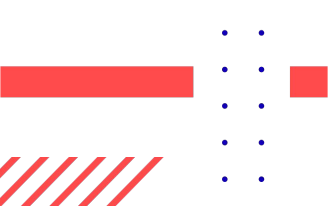
1. Flexbox



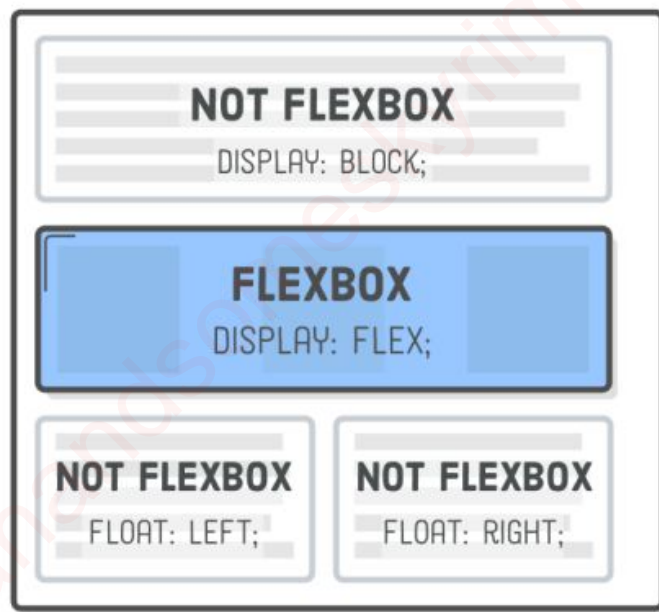
“FLEX CONTAINER”



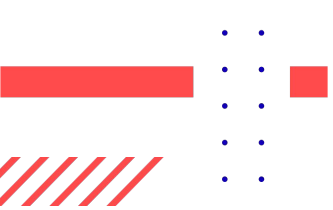
“FLEX ITEMS”



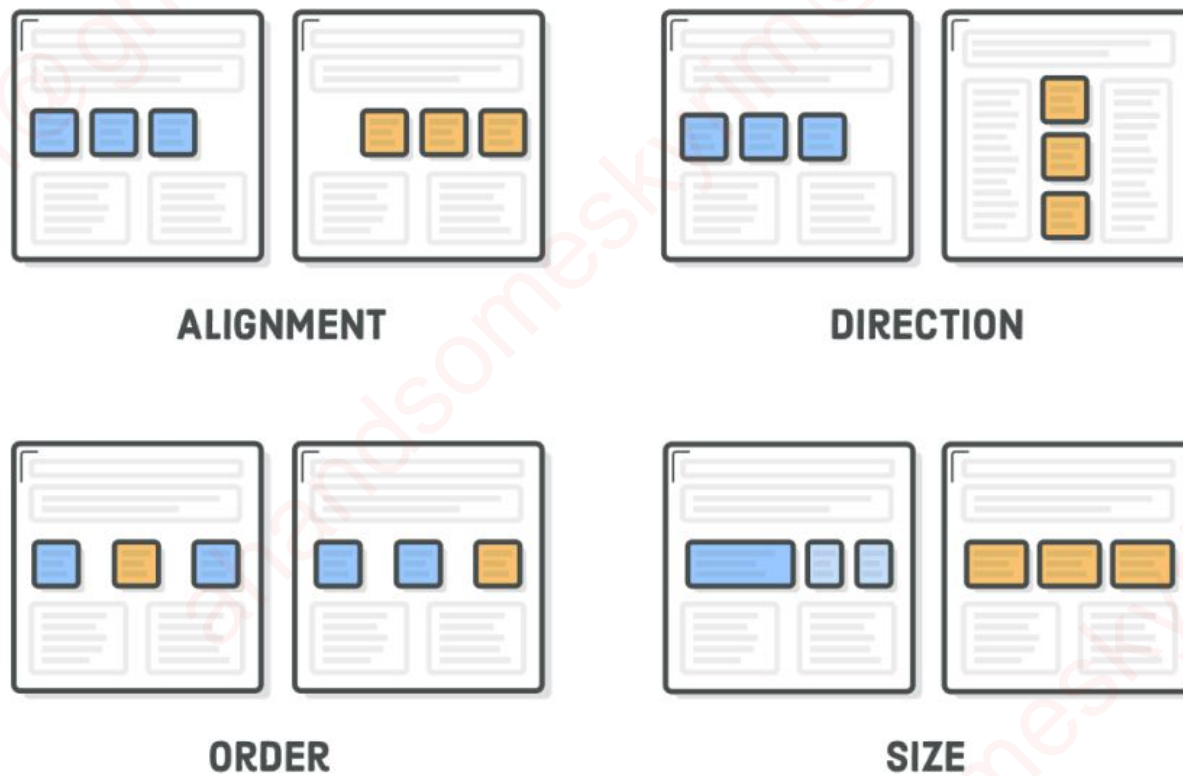
1. Flexbox

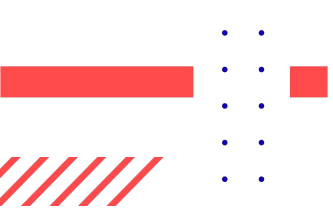


```
.container {  
  display: flex; /* or inline-flex */  
}
```

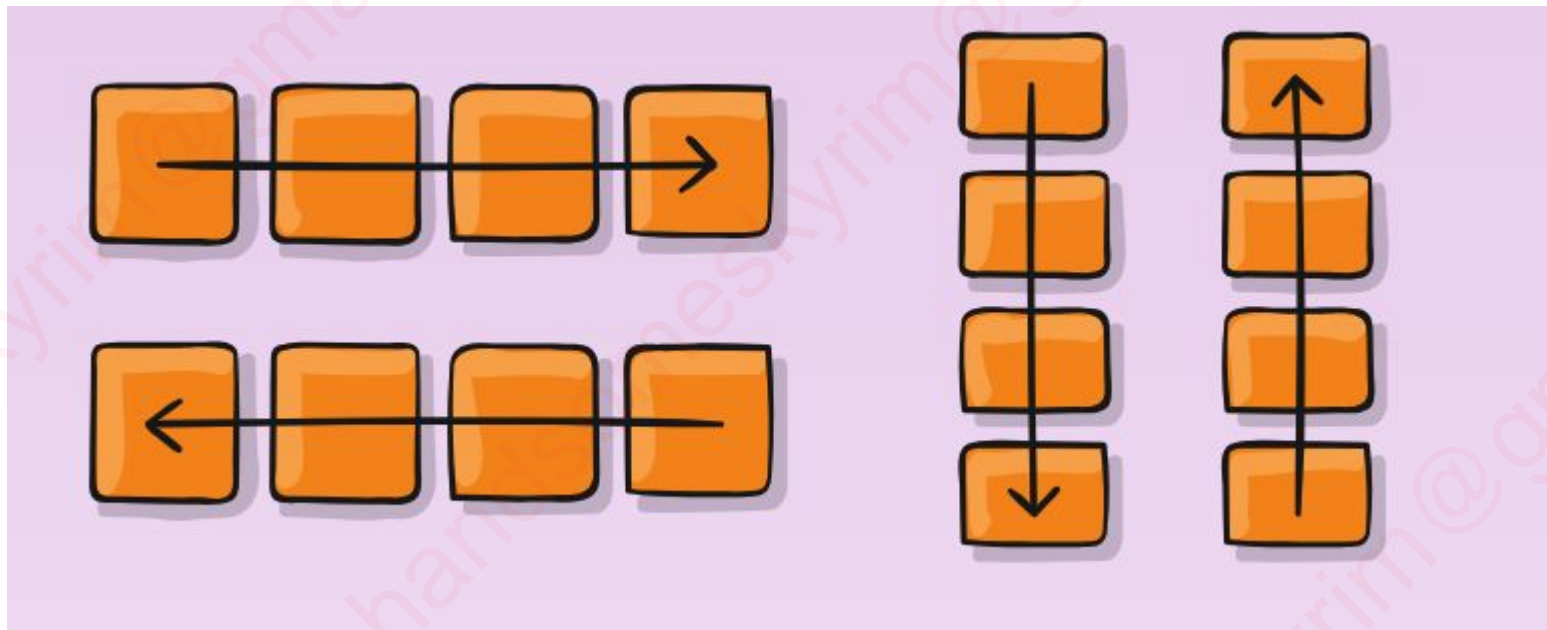


1. Flexbox

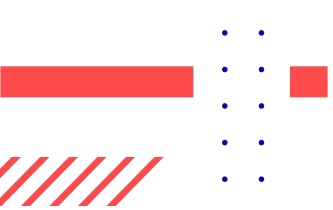




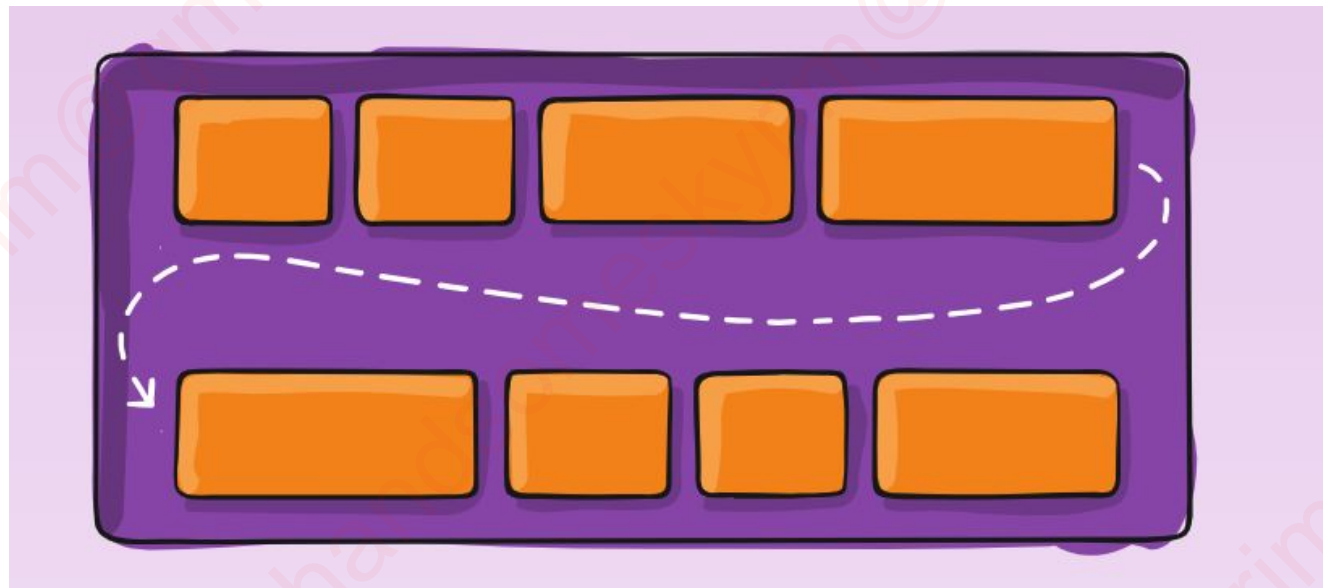
1. Flexbox



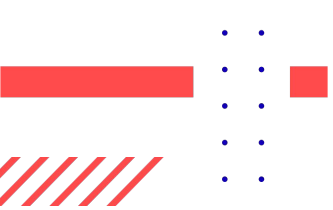
```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```

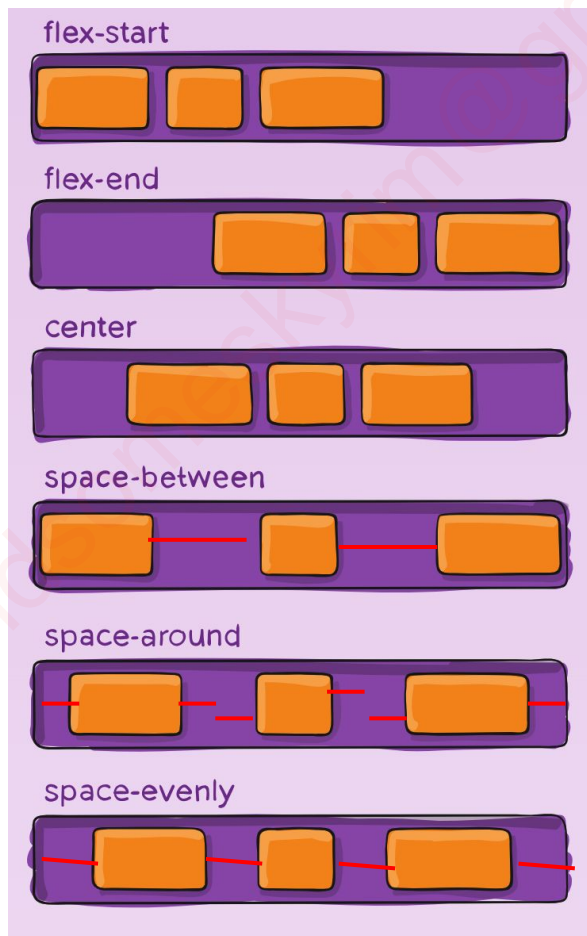
1. Flexbox



```
.container {  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

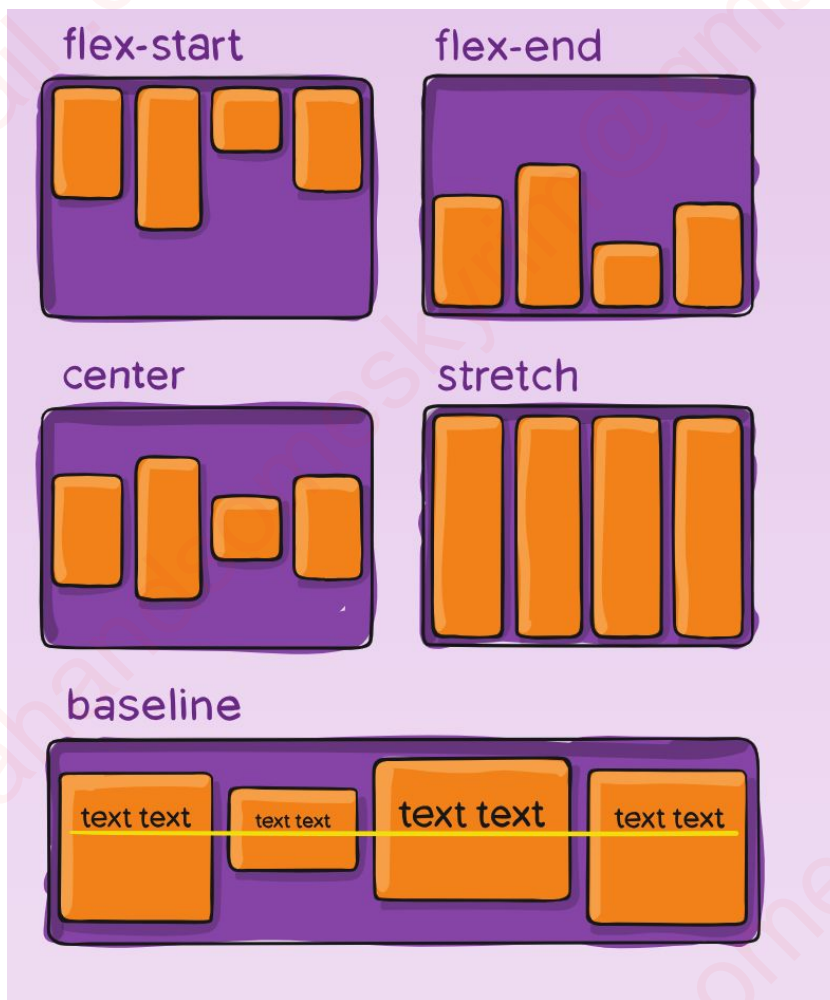


1. Flexbox



```
.container {  
  justify-content: flex-start | flex-end | center | space-between | space-around |  
  space-evenly | start | end | left | right ... + safe | unsafe;  
}
```

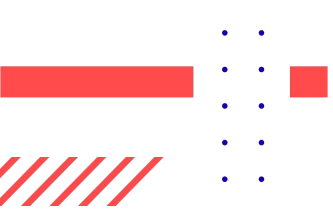
1. Flexbox



```
.container {
```

```
  align-items: stretch | flex-start | flex-end | center | baseline | first baseline | last baseline | start | end | self-start | self-end + ... safe | unsafe;
```

```
}
```

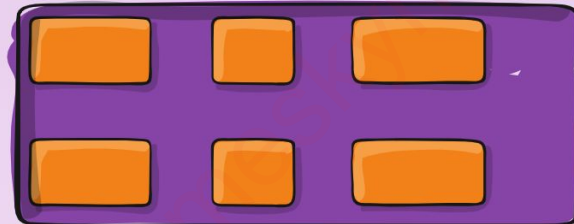


1. Flexbox

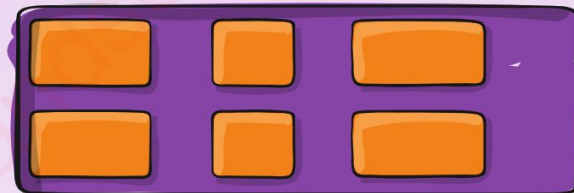
gap: 10px



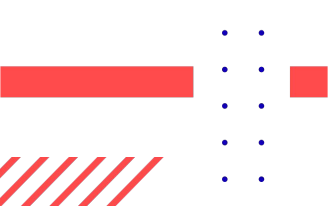
gap: 30px



gap: 10px 30px

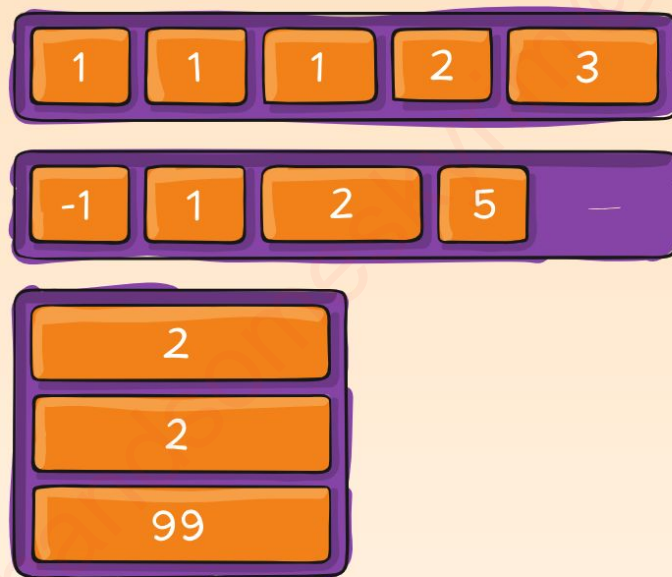


```
.container {  
  display: flex;  
  ...  
  gap: 10px;  
  gap: 10px 20px; /* row-gap column gap */  
  row-gap: 10px;  
  column-gap: 20px;  
}
```

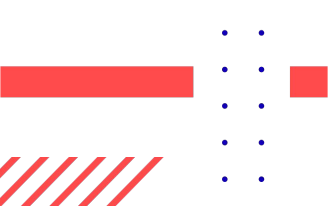


1. Flexbox

order



```
.item {  
  order: 5; /* default is 0 */  
}
```

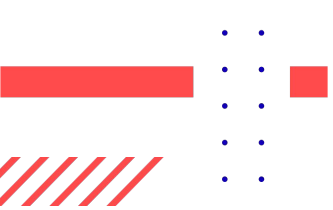


1. Flexbox

flex-grow



```
.item {  
  flex-grow: 4; /* default 0 */  
}
```



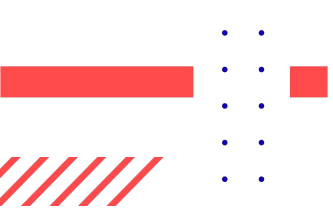
1. Flexbox

flex-grow



```
.item {  
  flex-grow: 4; /* default 0 */  
}
```

```
.item {  
  flex-shrink: 3; /* default 1 */  
}
```



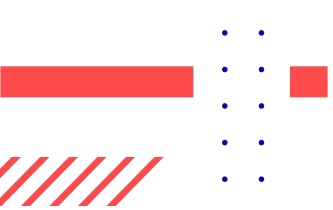
1. Flexbox

flex-grow



```
.item {  
  flex-grow: 4; /* default 0 */  
}
```

```
.item {  
  flex-shrink: 3; /* default 1 */  
}
```

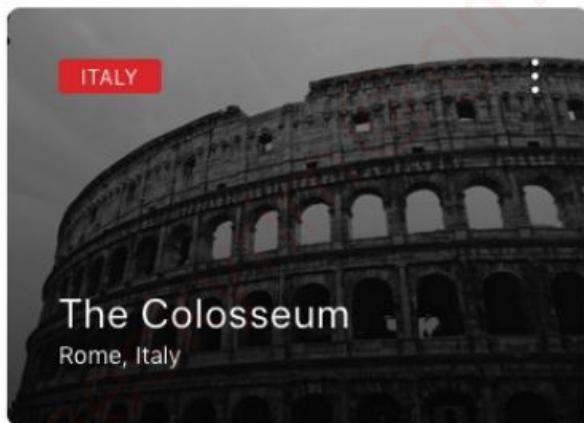



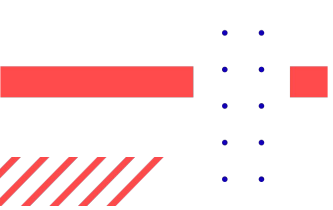
1. Flexbox

```
.item {  
  flex-basis: | auto; /* default auto */  
}
```

```
.item {  
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]  
}
```

1. Flexbox Practises





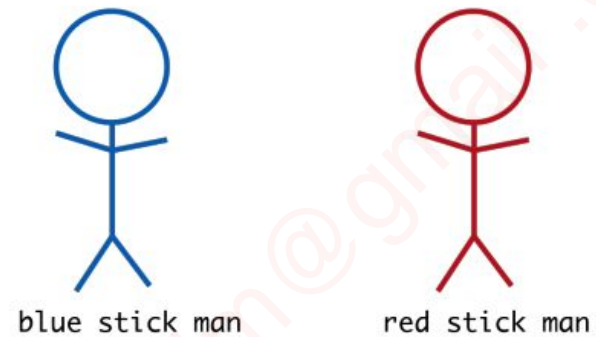
1. BEM



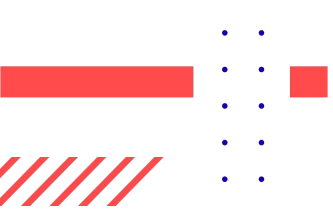
B for 'Block'



E for Elements



M for Modifiers



1. BEM

B for 'Block'

```
.stick-man {  
  }  
}
```

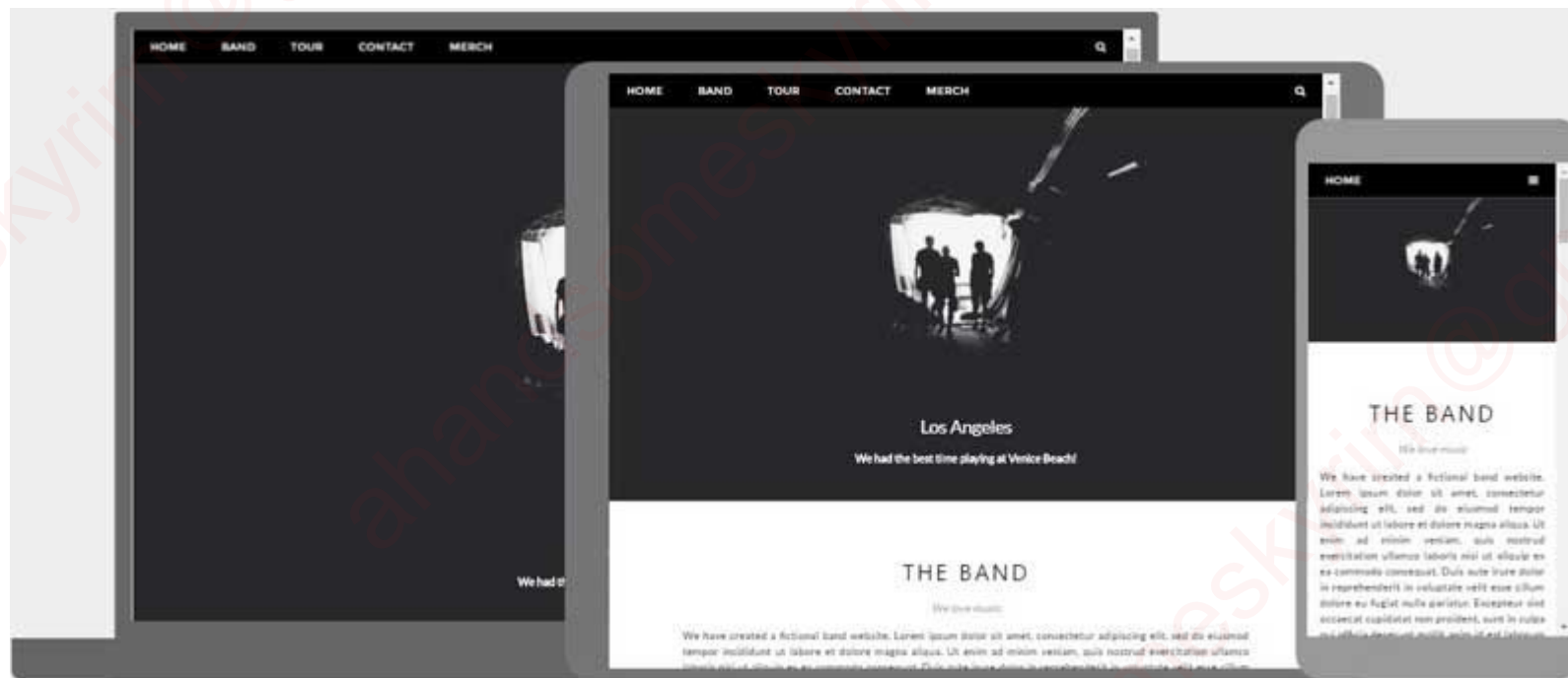
E for Elements

```
.stick-man__head {  
  }  
  
.stick-man__arms {  
  }  
  
.stick-man__feet {  
  }  
}
```

M for Modifiers

```
.stick-man--blue {  
  }  
  
.stick-man--red {  
  }  
}
```

2. Responsive Web Design



2. Responsive Web Design

Set the viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- Use the meta viewport tag to control the width and scaling of the browser's viewport.
- Include width=device-width to match the screen's width in device-independent pixels.
- Include initial-scale=1 to establish a 1:1 relationship between CSS pixels and device-independent pixels.
- Ensure your page is accessible by not disabling user scaling.

https://www.w3schools.com/html/example_withoutviewport.htm

https://www.w3schools.com/html/example_withviewport.htm

2. Responsive Web Design

Flexible Images

- `img {`
- `max-width:100%;`
- `}`

```
/* iPhone and Smartphones (portrait and landscape) */  
@media screen and (min-device-width : 320px) and (max-device-width: 480px) {  
    //你的样式放在这里...  
}
```

CSS3 Media Queries

- `@media screen and (max-width: 600px) {`
- `//你的样式放在这里....`
- `}`

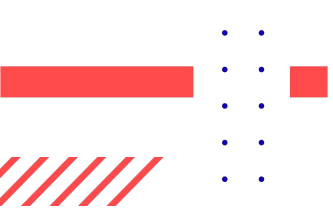
- `@media screen and (min-width: 900px) {`
- `//你的样式放在这里...`
- `}`

```

```

```

```



2. Responsive Web Design

Responsive Text Size

Responsive Web Page

Demo:

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive_media_query3

2. Responsive Web Design

| | Extra small <576px | Small ≥576px | Medium ≥768px | Large ≥992px | Extra large ≥1200px |
|---------------------|--------------------------------------|-----------------|------------------|-----------------|------------------------|
| Max container width | None (auto) | 540px | 720px | 960px | 1140px |
| Class prefix | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- |
| # of columns | 12 | | | | |
| Gutter width | 30px (15px on each side of a column) | | | | |
| Nestable | Yes | | | | |
| Column ordering | Yes | | | | |

```
<div class="row">
  <div class="col-sm-8">col-sm-8</div>
  <div class="col-sm-4">col-sm-4</div>
</div>
<div class="row">
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
</div>
```

3. Css Preprocessor



Less



Sass



PostCSS



1. CSS介绍

Selectors

A selector is part of a CSS rule that targets what HTML element it will be applying styles to. A selector can target an element or many elements at the same time.

简单选择器 (Simple selectors): 通过元素类型、class 或 id 匹配一个或多个元素。

```
h1 { }, .box { }, #header { }
```

属性选择器 (Attribute selectors): 通过 属性 / 属性值 匹配一个或多个元素。

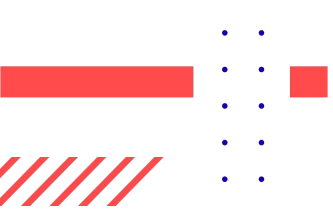
```
input[type="text"] { }, a[href="https://google.com"] { }
```

伪类选择器 (Pseudo-classes and pseudo-elements): 通过伪类或者伪元素

```
a:hover { }, p::first-line { }
```

组合选择器 (Combinators): 组合选择器

```
ul > li
```



1. CSS介绍

Simple Selectors

Element
Selector

```
body {  
    background-color:#cccc99;  
}
```

ID
Selector

```
#menu {  
    background-color:#ffff00;  
}
```

Class
Selector

```
.bookTitle {  
    font-style:italic;  
}
```

1. CSS介绍

Pseudo Selectors

伪类(Pseudo-classes): 匹配处于确定状态的一个或多个元素, 比如被鼠标指针悬停的元素, 或当前被选中或未选中的复选框, 或元素是DOM树中一父节点的第一个子节点。

```
a:hover {}  
a:focus {}  
div:first-child {}  
div:nth-child(2) {}  
div:last-child {}
```

1. CSS介绍

Pseudo Selectors

伪类(Pseudo-classes): 匹配处于确定状态的一个或多个元素, 比如被鼠标指针悬停的元素, 或当前被选中或未选中的复选框, 或元素是DOM树中一父节点的第一个子节点。

```
selector:pseudo-class {  
    property:value;  
}
```

```
/* unvisited link */ a:link {  
    color: #FF0000;  
}
```

```
/* visited link */ a:visited {  
    color: #00FF00;  
}
```

```
/* mouse over link */ a:hover {  
    color: #FF00FF;  
}
```

```
/* selected link */ a:active {  
    color: #0000FF;  
}
```

3. Css Preprocessor

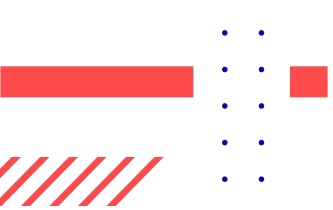
Sass

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
  
nav li {  
  display: inline-block;  
}  
  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```

<https://www.sassmeister.com/>

www.jiangren.com.au



3. Css Preprocessor

Sass (Syntactically Awesome Style Sheets) is a CSS preprocessor.

*.SCSS

Sass

Variables

```
$primaryColor: #eeffcc;

body {
  background: $primaryColor;
}
```

Match

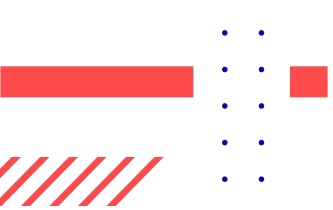
```
$container-width: 100%;

.container {
  width: $container-width;
}

.col-4 {
  width: $container-width / 4;
}
```

Nesting

```
a.myAnchor {
  color: blue;
  &:hover {
    text-decoration: underline;
  }
  &:visited {
    color: purple;
  }
}
```

3. Css Preprocessor

Sass (Syntactically Awesome Style Sheets) is a CSS preprocessor.

*.scss

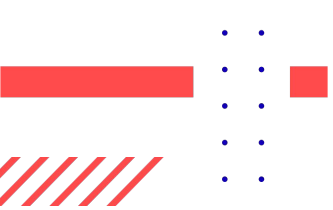
Sass

Imports

```
@import "grids";
```

Mixins

```
@mixin media($queryString){  
  
}
```



4. Design Principles

User familiarity

Consistency

Minimal surprise

4. Design Principles

Typography

- primary typeface,
- eventually secondary typeface (no need for a third one—two typefaces are usually enough, really),
- chosen font families (regular, italic, bold, semibold, light, ...),
- text styles examples (headings, paragraphs, info texts, ...).



Google Fonts

<https://fonts.google.com/>

Typefaces

Source Sans Pro

Source Sans Pro is an open-source sans serif typeface created for legibility in UI design. With a variety of weights that read easily at all sizes, Source Sans Pro provides clear headers as well as highly-readable body text.

Inspired by twentieth-century American gothic typeface design, its slender but open letters offer a clean and friendly simplicity. Advanced hinting allows Source Sans Pro to render well on Windows systems which run ClearType, and across browsers and devices. Moreover, it supports a variety of languages and alphabets, including Western and European language, Vietnamese, pinyin Romanization of Chinese, and Navajo.

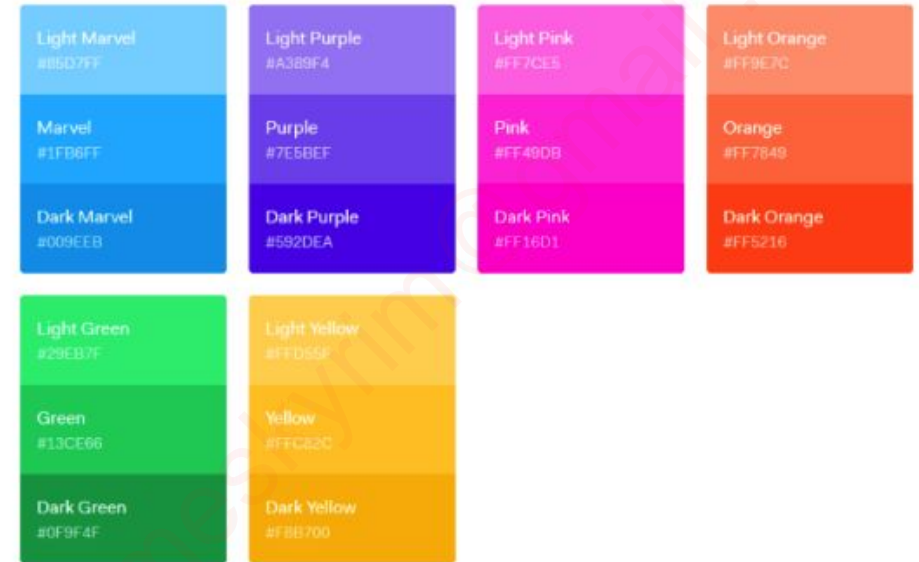
Aa

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789

4. Design Principles

Color Palette

- primary colours,
- secondary colours,
- complementary colours,
- alerts colours (warnings, errors, information validations, ...),
- additional colour elements (for example gradients or semi-transparent overlays).



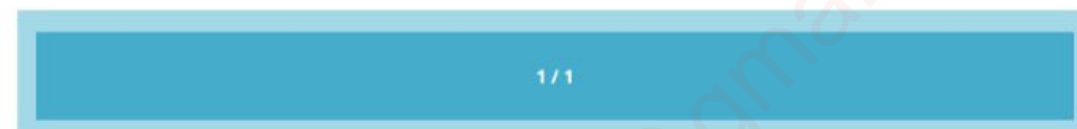
4. Design Principles

Spacing and Positioning

- grid system (sizes, gutters, nesting, ...),
- spacing rules and spacing patterns,
- spacing helpers (reusable constants for vertical and horizontal gutters between UI elements which can be used in the code).

Grid sizes

Size 1 of 1

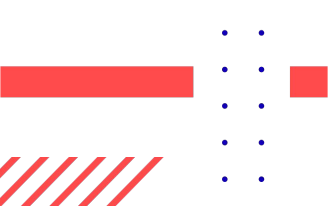


Size 1 of 2



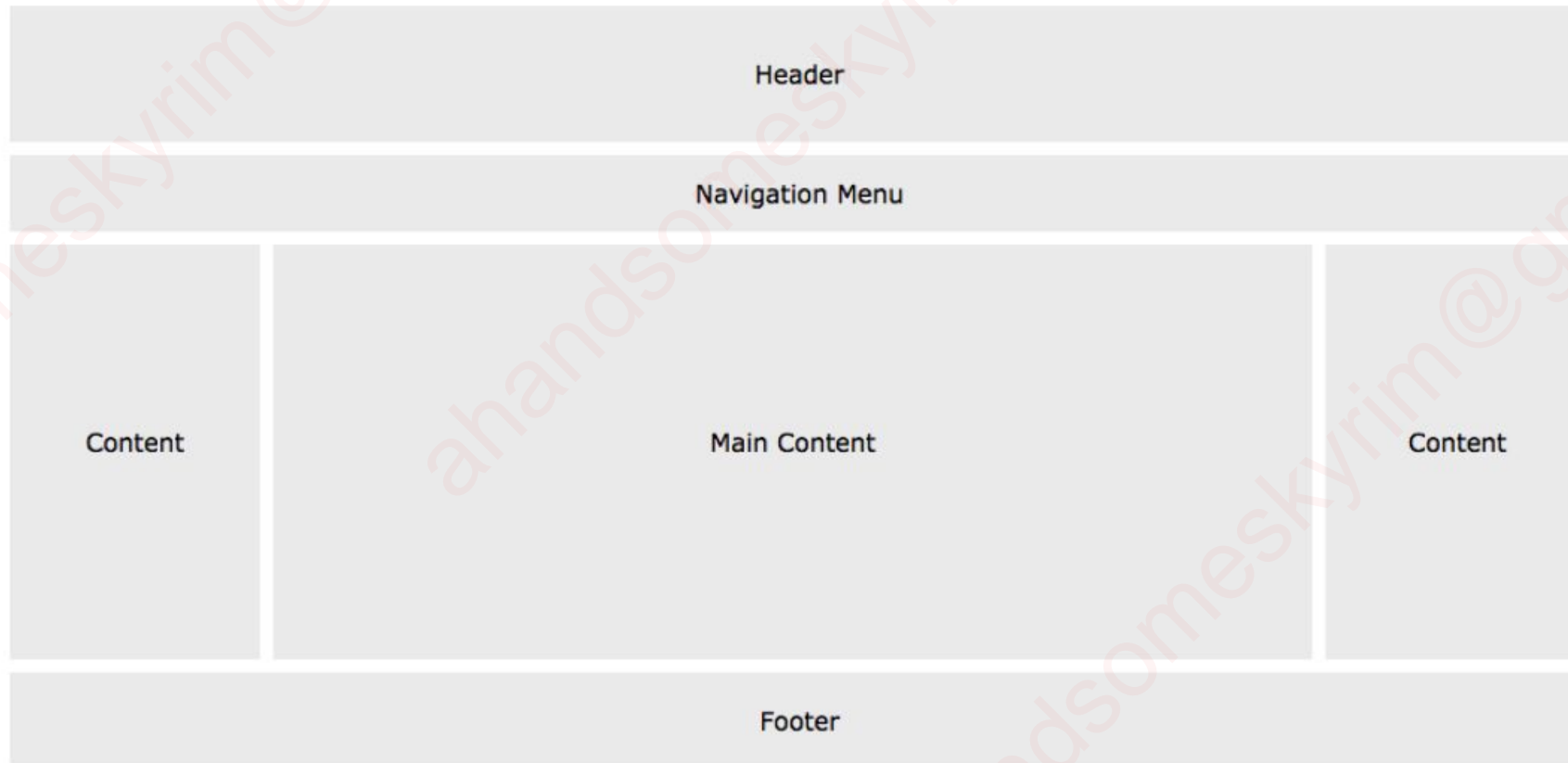
Size 1 of 3

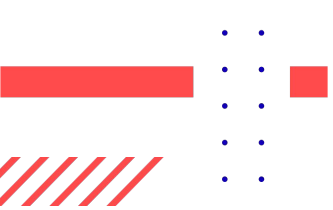




4. Design Principles

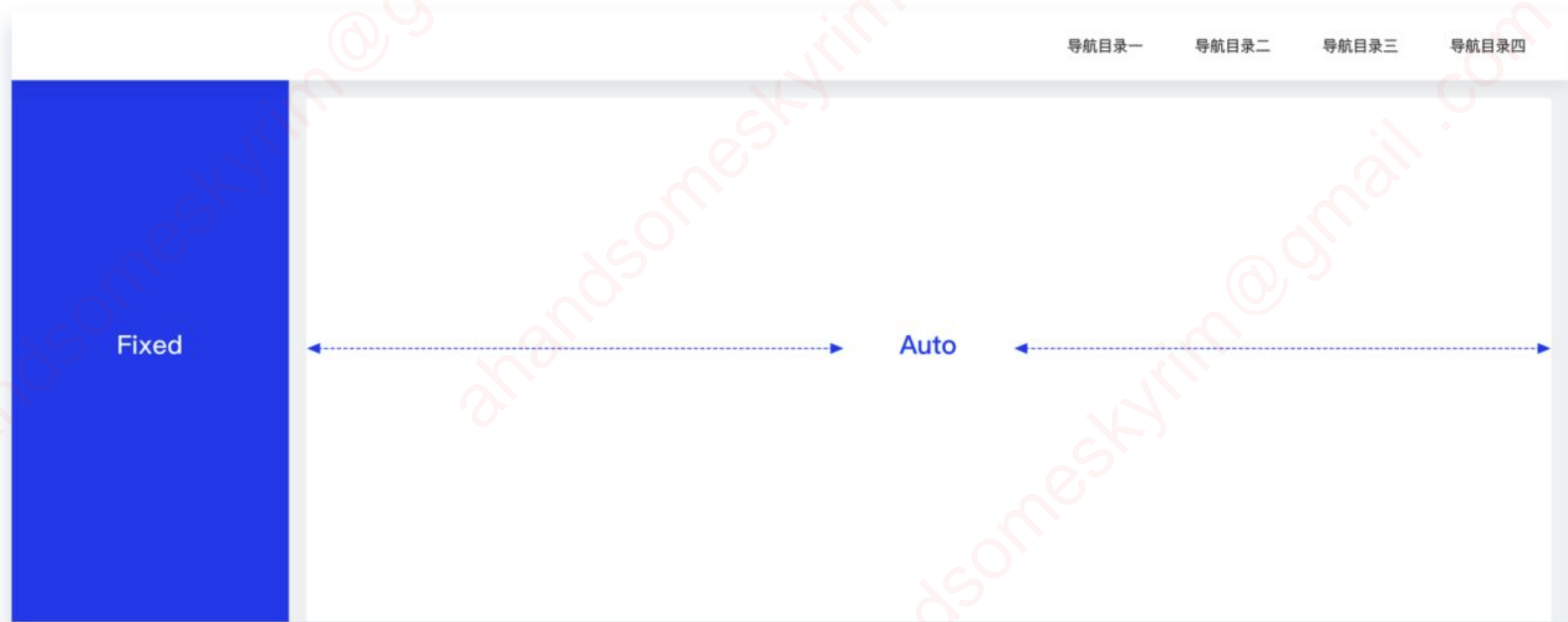
Layout

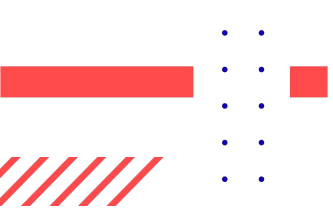




4. Design Principles

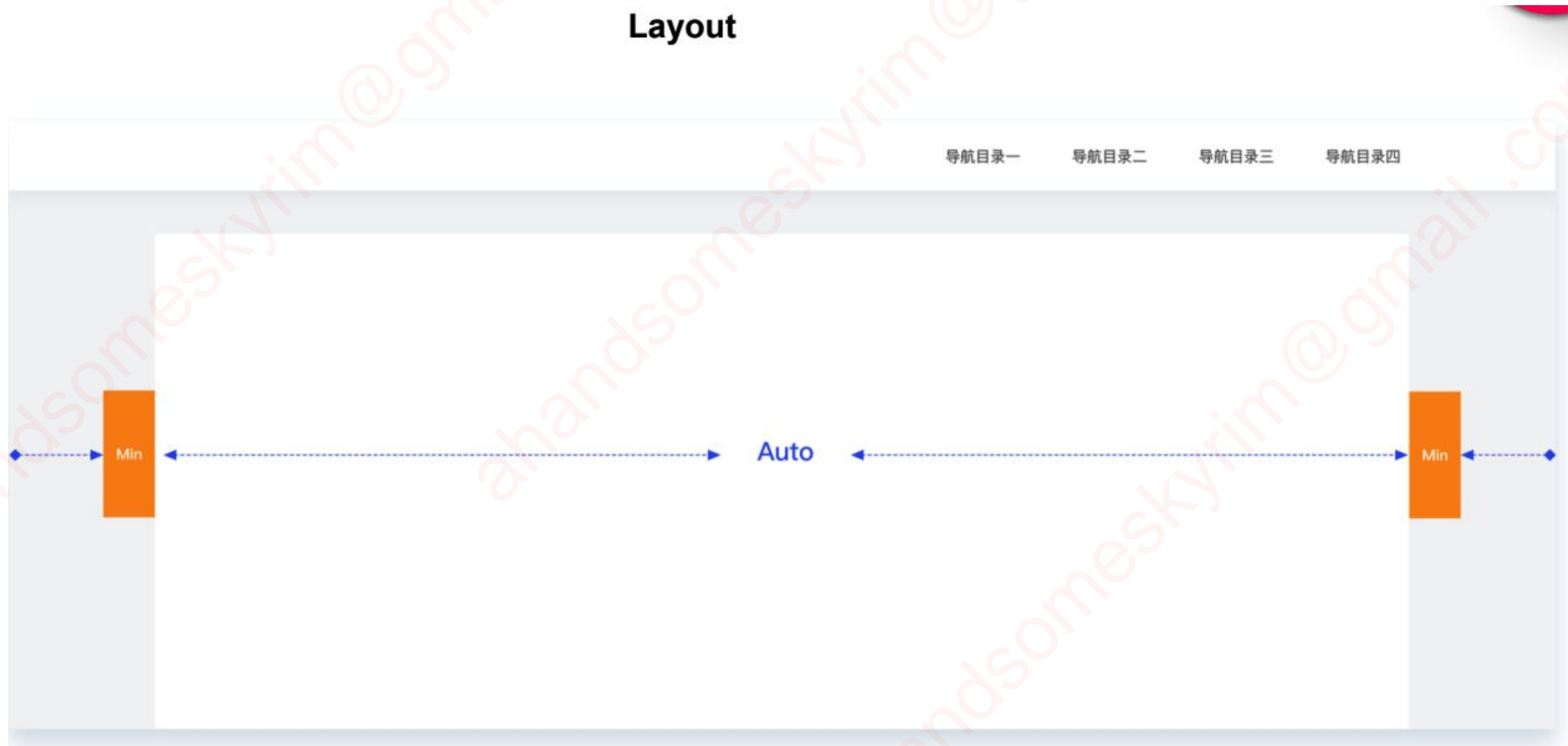
Layout

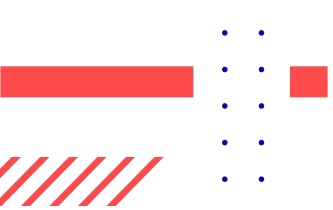




4. Design Principles

Layout

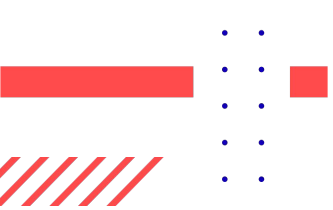




4. Design Principles

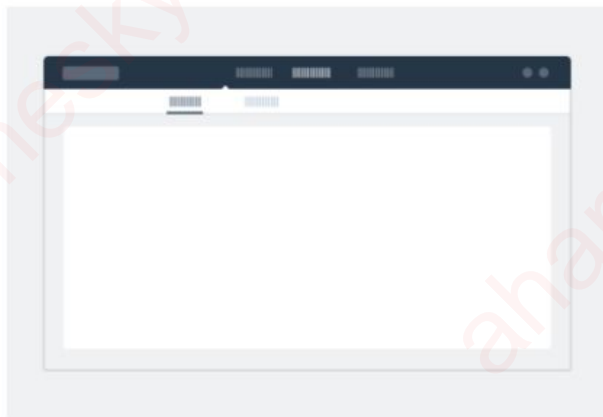
UI Elements

- navigation (main nav, secondary nav, breadcrumbs, pagination, ...),
- buttons (primary/secondary action buttons, tools buttons, ...),
- alert messages (warnings, errors, information, validations, ...),
- forms (fields, radio buttons, checkboxes, poplists, infield validation, ...),
- markers (person types markers, patient states markers, document types, ...),
- additional elements (calendar, patient timeline, patients lists, documents lists, cards, modal panels, ...).



4. Design Principles

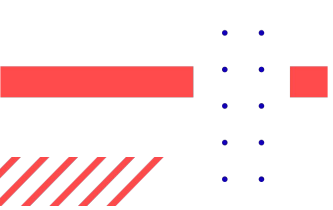
Navigation



Top Navigation

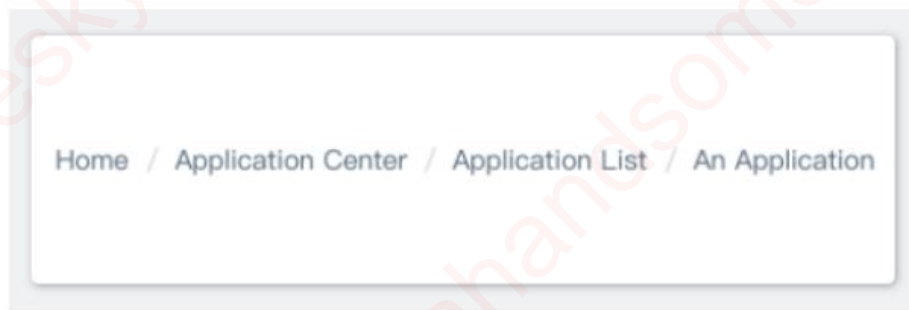


Side Navigation

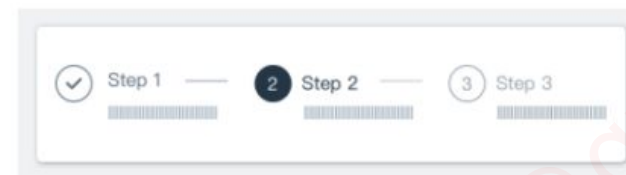


4. Design Principles

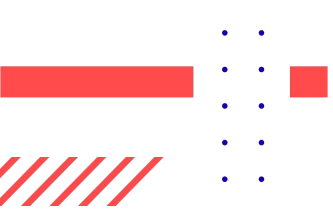
Navigation



Breadcrumb



Steps

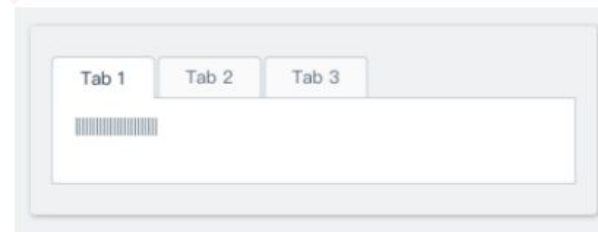


4. Design Principles

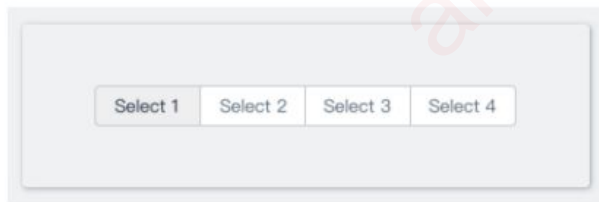
Navigation



Basic



Card

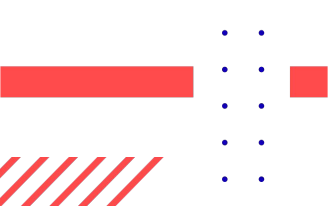


Pills



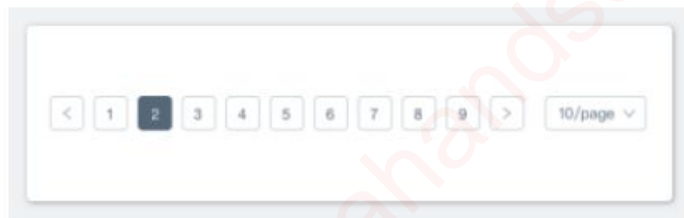
Vertical

Tabs



4. Design Principles

Navigation

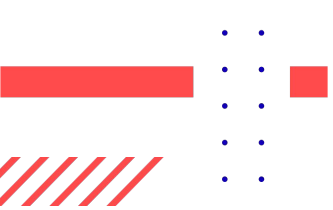


Pagination

VS Code Extension

Auto Close Tag
Auto Rename Tag
HTML Snippets
Babel ES6/ES7
Debugger for Chrome
ESLint
JavaScript (ES6) code snippets
Code Spell Checker
Node.js Modules Intellisense
React-Native/React/Redux snippets for ES6/ES7
TSLint
vscode-styled-components

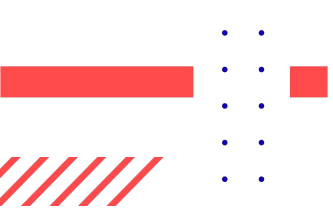
Color Highlight
Custom CSS and JS Loader
EditorConfig for VS Code
File Templates
Git History (git log)
Material Icon Theme
Prettier
Theme—Seti-Monokai



1. Flexbox

Quiz

https://docs.google.com/document/d/1f2HUO74wRMMISW_ROpw10nnZpYA7JdnEaW1NldzuQFw/edit?usp=sharing



1. Flexbox

Homework:

<https://flexboxfroggy.com/>



Q & A