



□ □ □ □ □ □ □ □

— — □ □ □ □ □ □

□ □ □ □ | □ □





□□□□□□



□□□□□□□□□□



□8□□□□/□□□□□□



□ □ □ □ □ ?

WEB APP HTML CSS JavaScript —

--	--	--	--	--	--	--	--	--

□□□□  
1990 - 2004

- `□□□□`
- `js□□`
- `□□□□□`

**web2.0**

**2004 - 2008**

- `□□□□□□□`
- `Ajax□□□□□□□`
- `□□`
- `□□:jQuery`




2008 - 2015

- V8□□□□
- Node.js□□
- MVVM/MVC
- /SPA□□□□□

□□□□□  
2015 - 2020

- 10개의 블록을 가지고 있다
- 10개의 블록을 가지고 있다

□□□□□  
2020 ~ ?

- 
- 
- 
- serverless
- lowcode/nocode
- WebAssembly

□ □ □ □ □ □ ?

The only constant in the world is change.

□□□□□□□□□□□□——□□□□□□□□□□□□ □□□.□□□

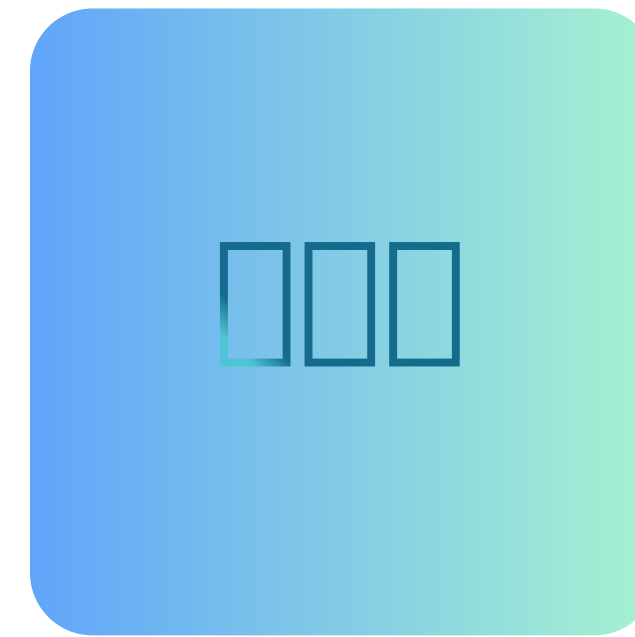
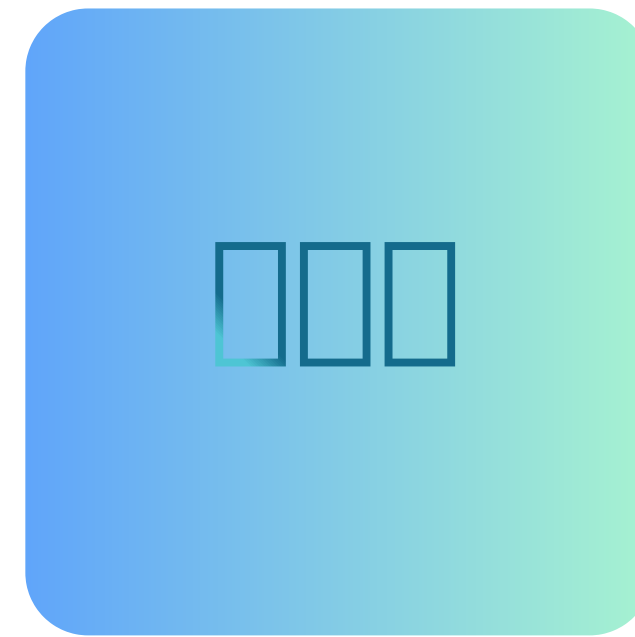
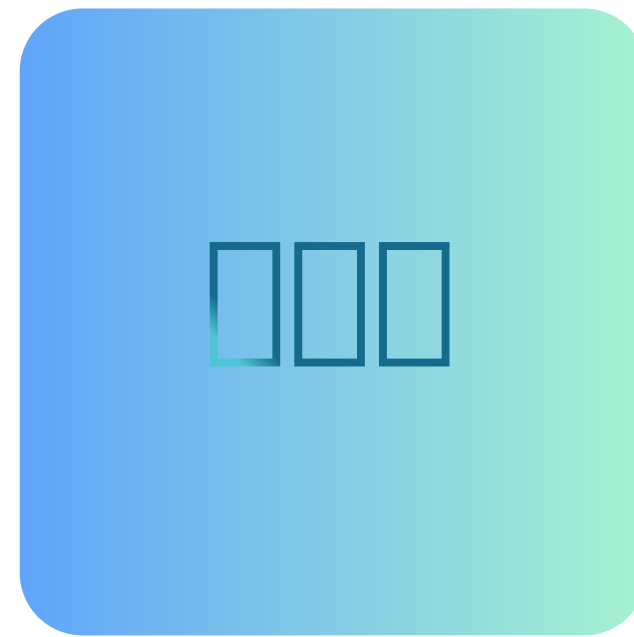
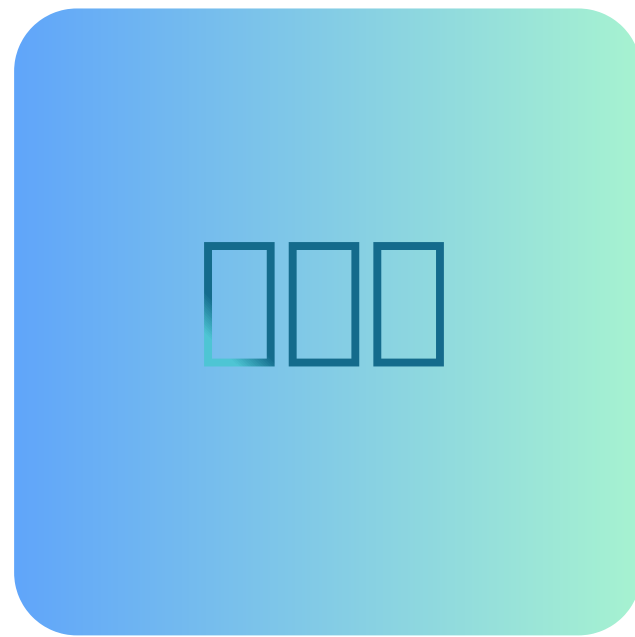
- `ReactDOM.render(

JS`
- `WebApp`
-

□ □ □ □ □ □ □ ?

[illegible]

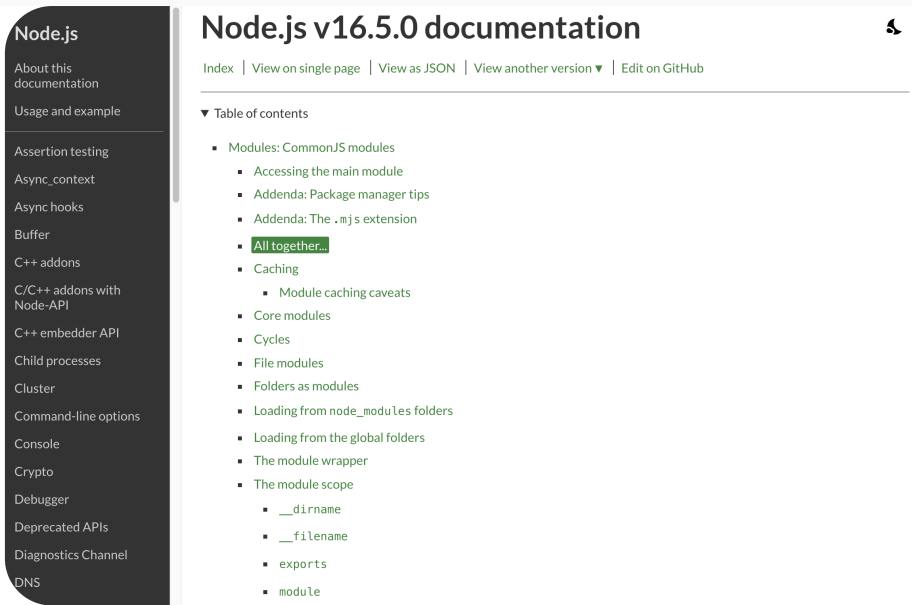
111



□ □

# 模块化-打包

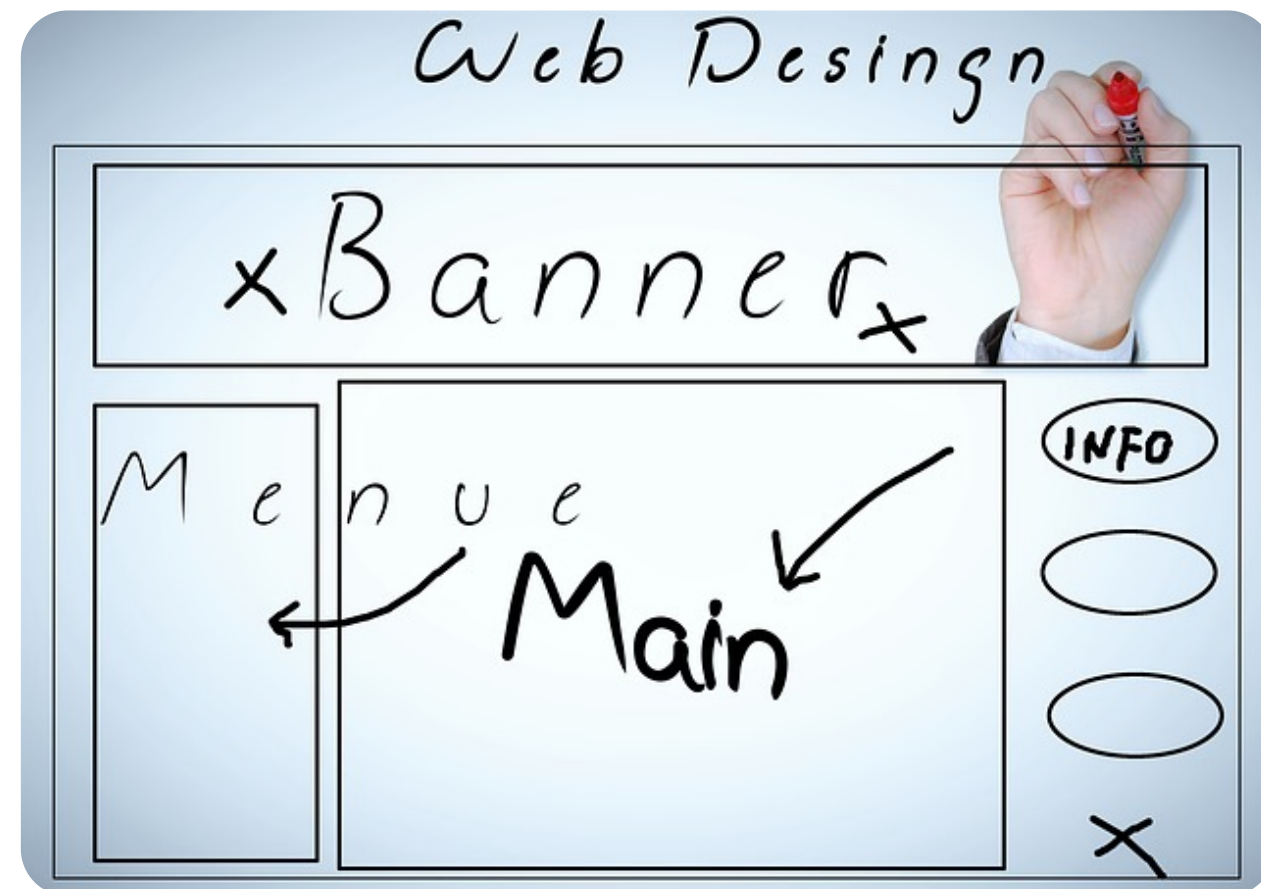
模块化-打包



- JS 打包方案CommonJS、AMD、CMD 打包 ES6 Module
- CSS 打包方案Sass、Less、Stylus、BEM、CSS Modules 打包方案BEM 打包方案CSS Modules 打包方案JS 打包方案打包方案JS 打包方案 CSS 打包方案 Vue 打包 style scoped
- 打包方案打包方案打包方案打包方案打包方案打包方案打包方案打包方案CSS打包方案打包方案JS 打包方案打包方案

□ UI □ □ □ □ □

- [illegible]



$$\boxed{\phantom{0}}\boxed{\phantom{0}}\boxed{\phantom{0}}\boxed{\phantom{0}}\boxed{\phantom{0}} - \boxed{\phantom{0}}\boxed{\phantom{0}}\boxed{\phantom{0}}$$

[illegible]

- 项目初始化;
- 项目代码规范制定(ESLint+StyleLint 等);
- 项目部署;
- 项目代码规范制定(BEM+Less+Stylus+CSS Modules 等);
- git flow 项目分支管理;
- code review;

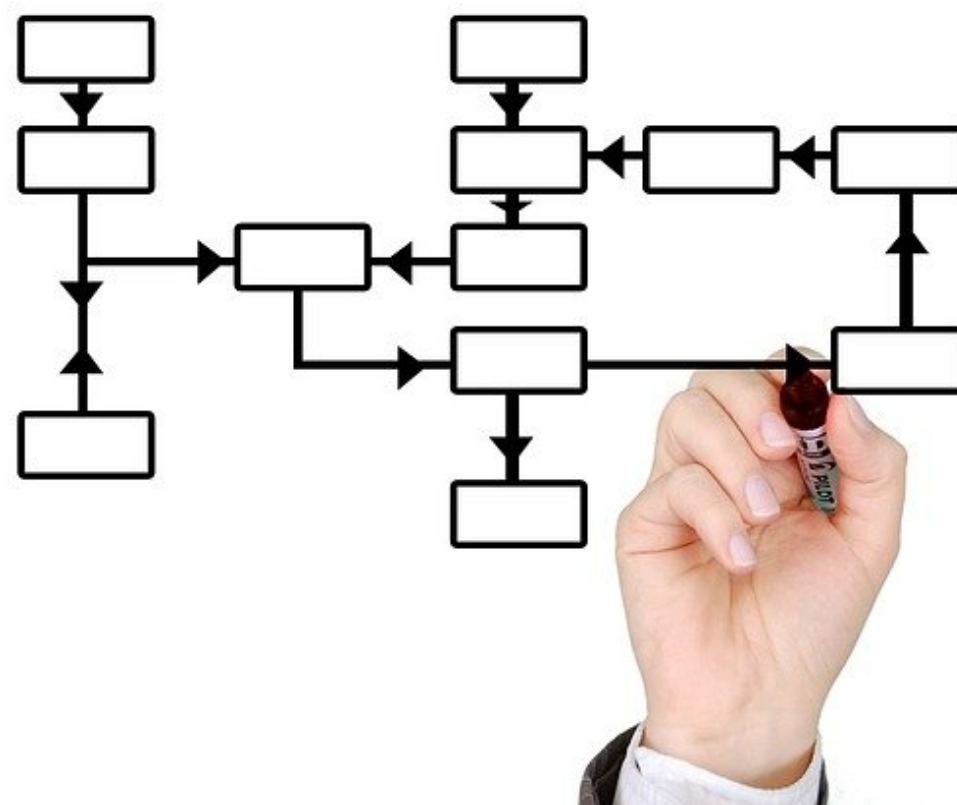
- ESLint & StyleLint

- □ □ □ ;

- `BEM``Sass``Less``Stylus``CSS Modules` `...`;

- git flow □□□□□□□□□□□□□□□□□□□□□□;

- code review;

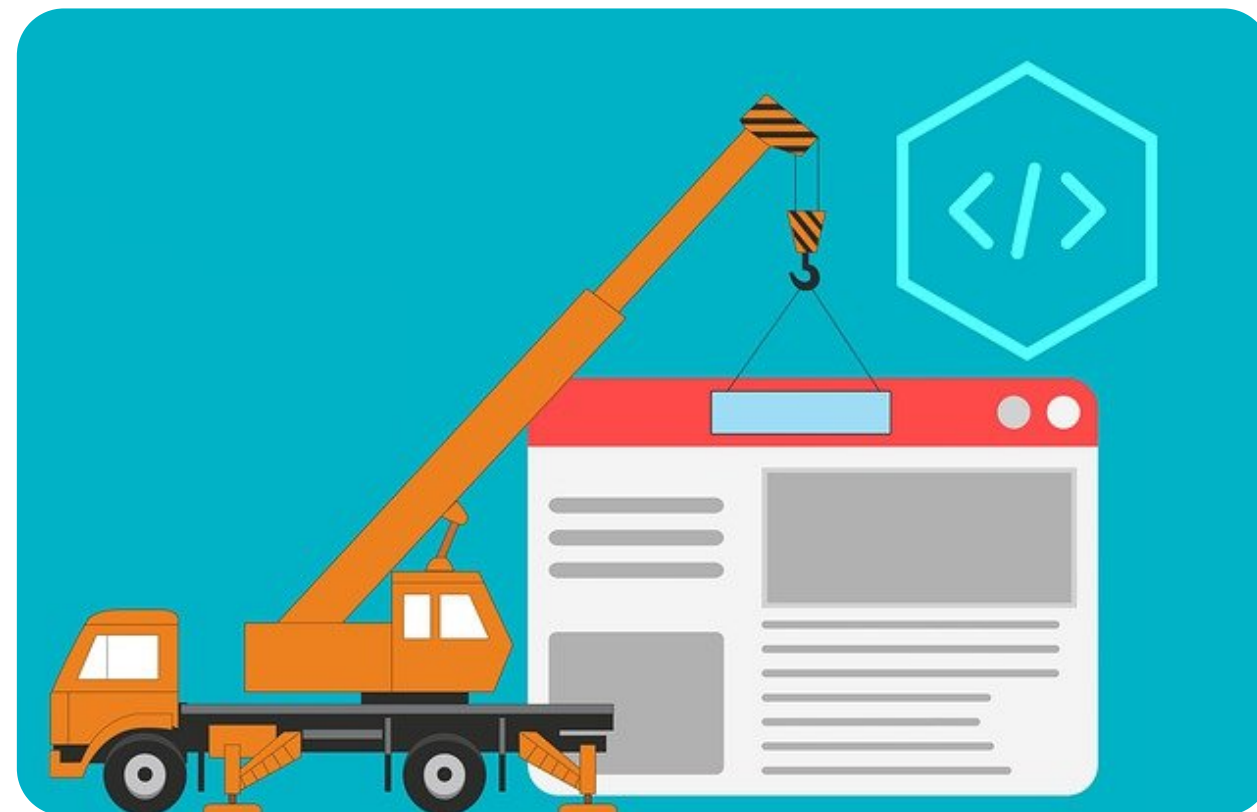


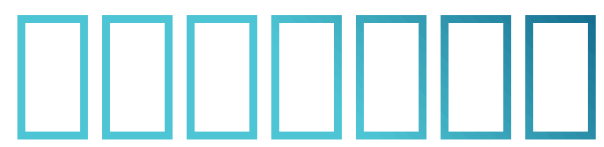


# 前端工程化-部署

前端工程化-部署

- webpack/parcel/ESbuild;
- CI/CD;
- 部署;

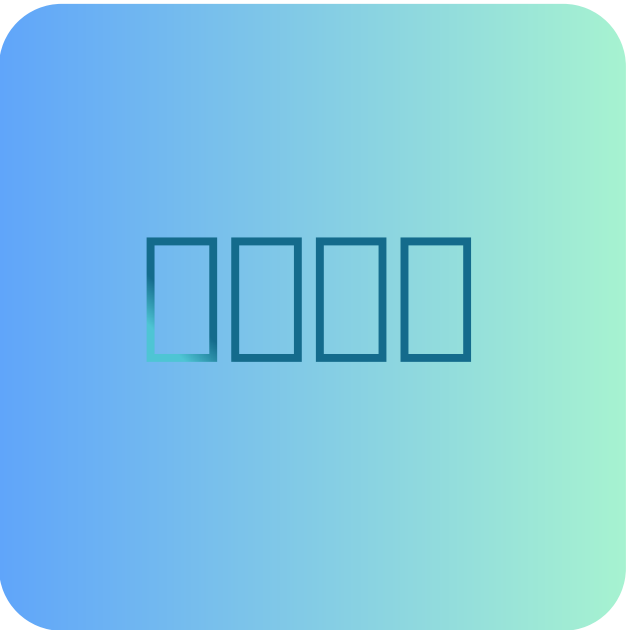
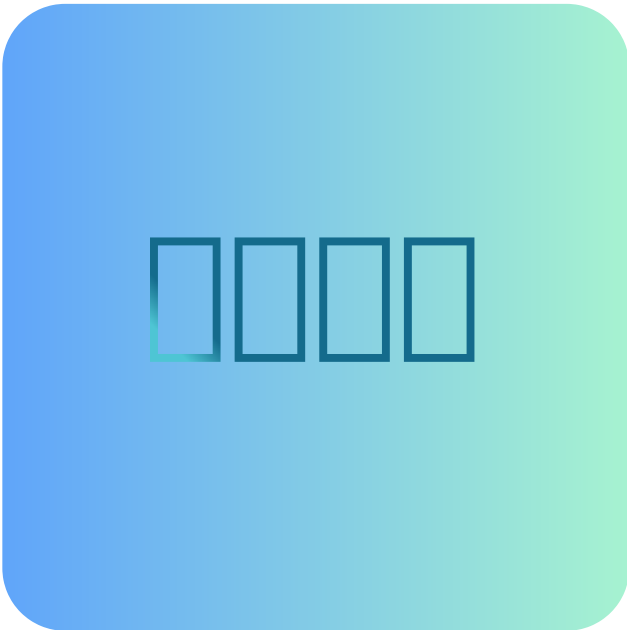
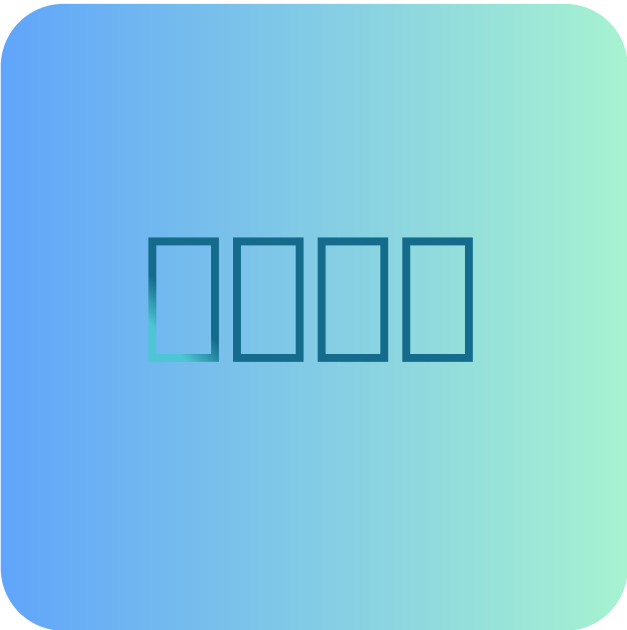
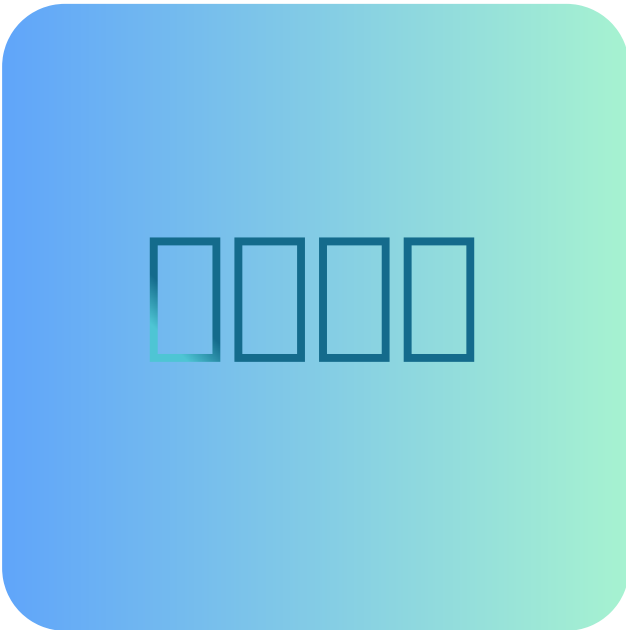




- □□□□□□□□
- □□×0×1×□□□□□□□□web□□□□□□



Well begun is half done.  
□□□□□□□□□□ — — □□□







□□□□□□□□VUE / REACT / ANGULAR / ...□?

	Angular	React	Vue
□□□□	2010□	2013□	2014
□□□□	TypeScript	□□□□JavaScript□□□	JavaScript/HTML/TypeScript
□□	Incremental DOM	Virtual DOM	Virtual DOM
□□□□	□□	□□	□□
□□	□□	□	□
□□□	□	□	□
□□□	□	□	□
□□□□	□□□□□□□□□□□□	□□Web□□SPA□□	□□□Web SPA□□

# Incremental DOM

Incremental DOM

Left

```
@Component({
  selector: 'todos-cmp',
  template: `
    <div *ngFor="let t of todos|async">
      {{t.description}}
    </div>
  `,
})
class TodosComponent {
  todos: Observable<Todo[]> = this.store.pipe(select('todos'))
  constructor(private store: Store<AppState>) {}
}
```

Right

```
var TodosComponent = /** @class */ (function () {
  function TodosComponent(store) { /* store component */ }
  TodosComponent.ngComponentDef = defineComponent({
    type: TodosComponent,
    selectors: [["todos-cmp"]],
    factory: function TodosComponent_Factory(t) {
      return new (t || TodosComponent)(directiveInject(Store));
    }, consts: 2, vars: 3,
    template: function TodosComponent_Template(rf, ctx) {
      if (rf & 1) { // create dom
        pipe(1, "async");
        template(0, TodosComponent_div_Template_0, 2, 1, null);
      } if (rf & 2) { // update dom
        elementProperty(0, "ngForOf", bind(pipeBind1(1, 1, ctx.todos)));
      }
    }, encapsulation: 2
  });
  return TodosComponent;
})();
```

# React - Virtual DOM

원래의 DOM은 브라우저에서 렌더링되는 DOM과 메모리에서 저장된 DOM이 일치한다.

## ORIGINAL DIV DOM

```
> var div = document.createElement('div')
var str = ''
for (var key in div){
  str += key + ' '
}
< "align title lang translate dir dataset hidden tabIndex accessKey draggable spellcheck contentEditable isContentEditable
offsetParent offsetTop offsetLeft offsetWidth offsetHeight style innerText outerText webkitDropzone onabort onblur
oncancel oncanplay oncanplaythrough onchange onclick onclose oncontextmenu oncuechange ondblclick ondrag ondragend
ondragenter ondragleave ondragover ondragstart ondrop ondurationchange onemptied onended onerror onfocus oninput
oninvalid onkeydown onkeypress onkeyup onload onloadeddata onloadedmetadata onloadstart onmousedown onmouseenter
onmouseleave onmousemove onmouseout onmouseover onmouseup onmousewheel onpause onplay onplaying onprogress onratechange
onreset onresize onscroll onseeked onseeking onselect onshow onstalled onsubmit onsuspend ontimeupdate ontoggle
onvolumechange onwaiting click focus blur onautocomplete onautocompleteerror namespaceURI prefix localName tagName id
className classList attributes innerHTML outerHTML shadowRoot scrollTop scrollLeft scrollWidth scrollHeight clientTop
clientLeft clientWidth clientHeight onbeforecopy onbeforecut onbeforepaste oncopy oncut onpaste onsearch onselectstart
onwheel onwebkitfullscreenchange onwebkitfullscreenerror previousElementSibling nextElementSibling children
firstElementChild lastElementChild childElementCount hasAttributes getAttribute getAttributeNS setAttribute
setAttributeNS removeAttribute removeAttributeNS hasAttribute hasAttributeNS getAttributeNode getAttributeNodeNS
setAttributeNode setAttributeNodeNS removeAttributeNode closest matches webkitMatchesSelector getElementsByTagName
getElementsByName getElementsByClassName insertAdjacentElement insertAdjacentText insertAdjacentHTML
createShadowRoot getDestinationInsertionPoints requestPointerLock getClientRects getBoundingClientRect scrollIntoView
scrollIntoViewIfNeeded animate remove webkitRequestFullscreen webkitRequestFullscreen querySelector querySelectorAll
ELEMENT_NODE ATTRIBUTE_NODE TEXT_NODE CDATA_SECTION_NODE ENTITY_REFERENCE_NODE ENTITY_NODE PROCESSING_INSTRUCTION_NODE
COMMENT_NODE DOCUMENT_NODE DOCUMENT_TYPE_NODE DOCUMENT_FRAGMENT_NODE NOTATION_NODE DOCUMENT_POSITION_DISCONNECTED
DOCUMENT_POSITION_PRECEDING DOCUMENT_POSITION_FOLLOWING DOCUMENT_POSITION_CONTAINS DOCUMENT_POSITION_CONTAINED_BY
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC nodeType nodeName baseURI isConnected ownerDocument parentNode parentElement
childNodes firstChild lastChild previousSibling nextSibling nodeValue textContent hasChildNodes normalize cloneNode
isEqualNode isSameNode compareDocumentPosition contains lookupPrefix lookupNamespaceURI isDefaultNamespace insertBefore
appendChild replaceChild removeChild addEventListener removeEventListener dispatchEvent "
```

React:

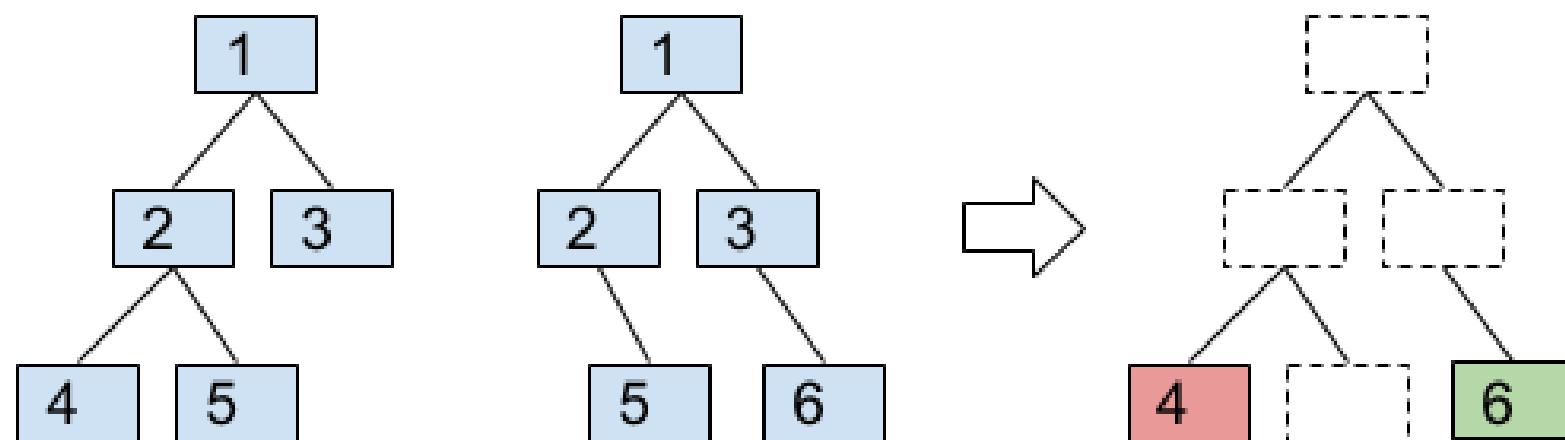


React는 DOM을 메모리에서 렌더링하는 Virtual DOM을 사용하여 web 브라우저와 ReactNative, Weex.



React는 DOM을 메모리에서 렌더링하는 Virtual DOM을 사용하여 web 브라우저와 ReactNative, Weex.

## Virtual DOM DIFF





# □□□□ - React or Vue ?

# ○○○○○○○○JavaScript ○○○ – ○○○○○

□ □ □ □

# jQuery

Page 10

□ □ □ □

□ □ □ □

111

□□□□□/□□□□

5/5/2020

11

□ □ □ □

# Github

Page 10

Page 10

□ □ □ □

Angular

# 📦📦📦📦 - 📦📦📦📦

Talk is cheap. Show me the code. — Linus Torvalds

## 1. 📦📦📦📦 Vue Cli

```
npm install -g @vue/cli
```

## 2. 📦📦📦📦

```
vue create fintech-financeRecon-portal
```

- 📦📦📦📦📦📦nodejs📦📦npm📦📦📦📦📦📦📦npm📦📦📦📦📦📦📦📦📦📦

# 📁📁📁📁 - 📁📁📁📁

convention over configuration 📁📁📁📁

## ■ 📁📁📁📁

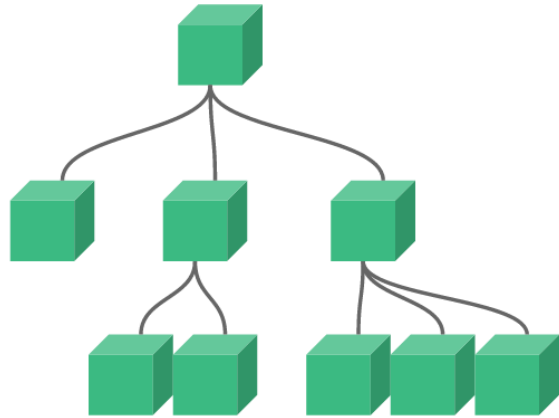
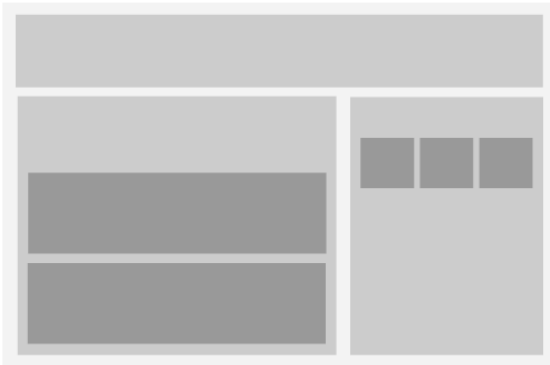
```
├─ README.md      // 📄 📄 📄
├─ package.json   // 📄 📄 📄 📄
├─ src
│   ├─ App.vue    // 📄 📄 📄
│   ├─ api        // api 📄
│   ├─ assets     // 📄 📄 📄
│   ├─ components // 📄 📄 📄 📄
│   │   └─ Charts
│   ├─ config     // 📄 📄 📄 📄
│   │   └─ defaultSettings.js
│   ├─ main.js    // js 📄
│   ├─ router.js  // 📄 📄 📄
│   ├─ store      // 📄 📄 📄
│   ├─ utils      // 📄
│   │   └─ axios.js
│   │   └─ utils.js
│   └─ views      // 📄
│       └─ 404.vue
└─ .env.dev       // 📄 📄 📄 📄
```



# UI - UX

UI/UX Design Process

- UI/UX Design



- UI/UX Design for fintech applications

+---+-----+		
	hbTablePage	
+---+-----+		
M	search-bar	
E	action-bar	
N	data-table	
U		
+---+-----+		

UI/UX Design Process

UI/UX Design Process Vue



$$\begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} - \begin{array}{|c|c|c|c|} \hline & & & \\ \hline \end{array} \begin{array}{|c|} \hline \times \\ \hline \end{array} 2$$

[illegible]

Table 1 - Summary

☐ \_\_\_\_\_

```
/**
 * TablePage.vue
 * 数据表格
 * this.$emit('myEvent')
 */
onSelectChange(selectedRowKeys, selectedRows) {
  this.$emit('ON_SELECTED_DATA', this.selectedData);
}
```

■ ☐ ☐ ☐ ☐ ☐

```
/* Page.vue */
<template>
  <div>
    <hbTablePage
      @ON_SELECTED_DATA="onSelectedData"
    >
    </hbTablePage>
  </div>
</template>
```

□ □ □ □

- `prop` “`prop`” 属性名を指定する;
- `update:myPropName` 更新する属性名



# Table - 3

Table component with slots for search bar and action bar

## Table component - Slots

### Table component slots

```
/* TablePage.vue */
<div v-if="$slots.searchBar">
  <slot name="searchBar"></slot>
  <a-divider dashed style="margin: 12px 0px" />
</div>
<div v-if="$slots.actionBar">
  <slot name="actionBar"></slot>
</div>
```

### Table component

```
/* Page.vue */
<template>
  <div>
    <hbTablePage>
      <template v-slot:searchBar>
        <hbTableSearchBar />
      </template>
      <template v-slot:actionBar>
        <hbTableActionBar />
      </template>
    </hbTablePage>
  </div>
</template>
```

# 路由 - 路由

路由配置

## 路由配置

```
/* router.js
* 路由配置 */
{
  path: '/userLogin',
  name: 'userLogin',
  component: resolve => require(['@/views/user/Login'],
    resolve),
}
```

```
/* router.js
* 路由配置 */
{
  path: '/userLogin',
  name: 'userLogin',
  component: =>import(
    /* webpackChunkName: "userLogin" */
    '../views/userLogin.vue')
}
```

## 路由配置

```
/* babel-plugin-component
* .babelrc */
{
  "presets": [["es2015", { "modules": false }]],
  "plugins": [
    "component",
    {"libraryName": "ant-design-vue",
      "styleLibraryName": "theme-chalk"}]
  ]
}
```

```
/* main.js or page.vue */
import Vue from 'vue';
import { Button, message } from 'ant-design-vue';

Vue.use(Button);
Vue.prototype.$message = message;
```

# webpack - 1

webpack 配置

## ■ 配置

```
/* image-webpack-loader
 * vue.config.js */
{
  test: /\. (png|jpe?g|gif|svg)(\?.*)?$/,
  use: [{
    loader: 'url-loader',
    options: {
      limit: 10000,
      name: utils.assetsPath('img/[name].[hash:7].[ext]')
    }
  }, {
    loader: 'image-webpack-loader',
    options: {
      bypassOnDebug: true,
    }
  }]
}
```

## ■ 配置

```
/* CommonsChunkPlugin
 * vue.config.js
 * package.json 配置 vendor.js 文件 */
new webpack.optimize.CommonsChunkPlugin({
  name: 'vendor',
  minChunks: function(module, count) {
    return (
      module.resource &&
      /\.js$/ .test(module.resource) &&
      module.resource.indexOf(
        path.join(__dirname, '../node_modules')
      ) === 0
    );
  }
}),
/* 配置 */
new webpack.optimize.CommonsChunkPlugin({
  name: 'manifest',
  chunks: ['vendor']
})
```

# webpack - 2

webpack 配置

## ■ ES6 转 ES5 配置

```
/* babel-plugin-transform-runtime
* .babelrc */
"plugins": [
  "transform-runtime"
]
```

## ■ BABEL 转 ES6 转 ES5 配置

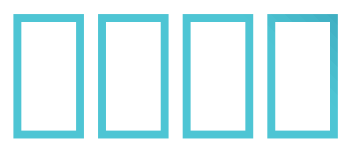
## ■ 转 ES6 转 ES5 配置

## ■ gzip 配置

```
/* compression-webpack-plugin@5.0.1
* vue.config.js
* 配置 .gz 文件 */
const CompressionPlugin = require("compression-webpack-plugin")
configureWebpack: {
  plugins: [
    new CompressionPlugin({
      filename: "[path].gz",
      algorithm: "gzip",
      test: /\.js$|\.css$|\.html$/,
      threshold: 10240, // 文件大于 10240 字节才会被压缩
      minRatio: 0.8,
      deleteOriginalAssets: false
    })
  ]
}
```

```
/* nginx.conf
* nginx 配置 gzip */
gzip on; # 配置 gzip
...
```





□□□□□□□□

- □□□□□□□□□□×0×1□□□□□□□□

- □□□□:

1. □□vue cli□□□□□□□□□□

2. □□□□□□□□□□□□□□□□

3. □□□□□□□□□□□□□□□□□□□□;

□□VueJs□□□□□□□□□ <https://vuejs.org/>.

# Where to Go From Here?

Flutter 1.10.0

Flutter 1.10.0...  
Flutter 1.10.0...

Flutter

Flutter

Flutter

Flutter

Flutter

Flutter

Flutter

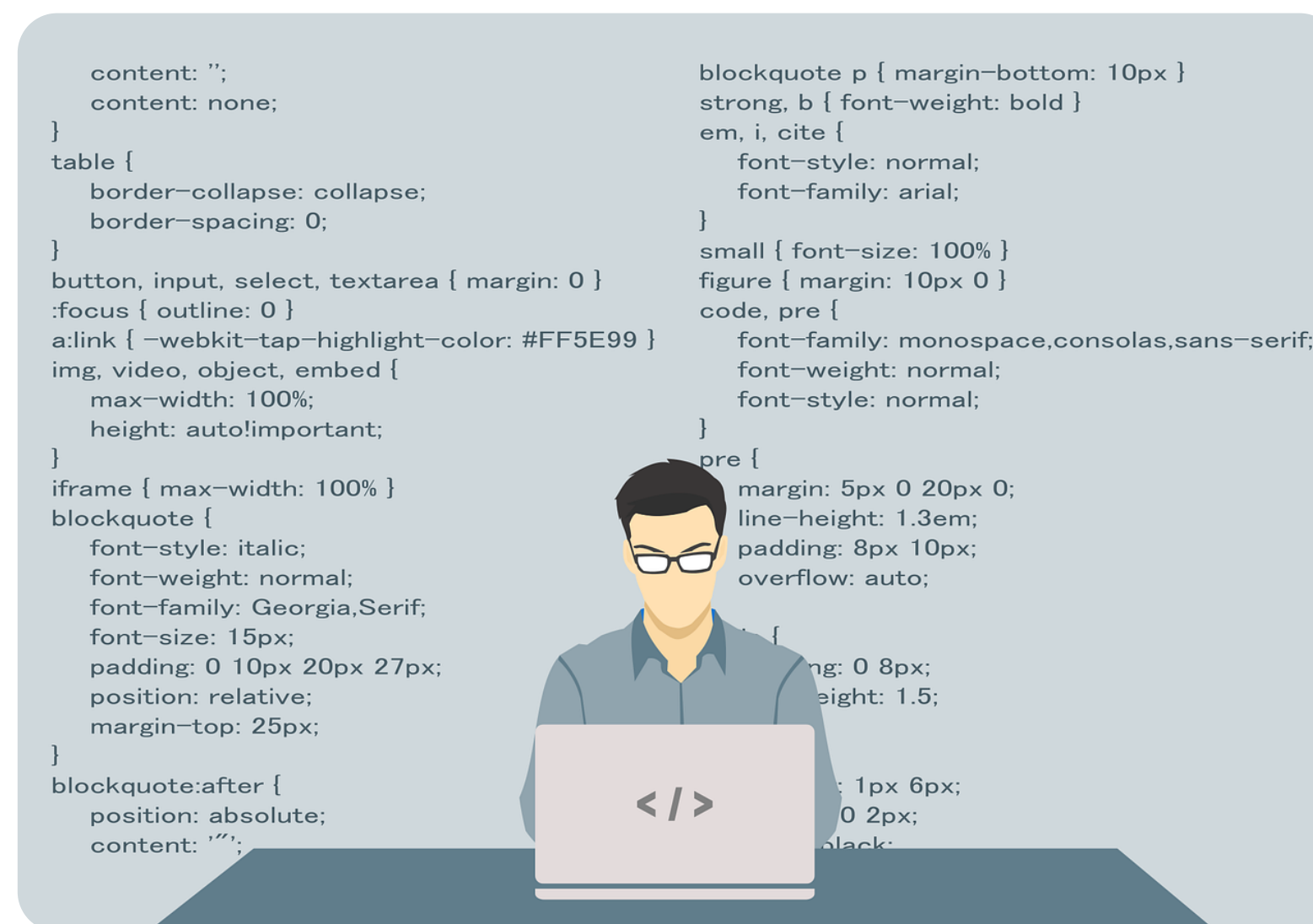
.....

# 前端技术栈

- 基础库 (jQuery, ES5, ES6)
- 框架 (reactjs, vuejs, angularjs) / jQuery 生态
- SPA 单页应用



- `0vuejsweb`;
- `vuejs`



THANK YOU !

□□□□□□□□□□□□

□□□□ □ □□



# One More Thing

□□□□□□□□□□<https://sli.dev>