查看 系统内存 ：

cmd ---> dxdiag

查看Chrome版本 :

chrome://version/

**========================================================**

Js \ Jq

window.innerWidth能获取当前窗口的宽度(包含滚动条), 在IE8浏览器是无效的

$(window).width()获取当前窗口的宽度(不包含滚动条)

通过代码使用媒体查询 :

if (window.matchMedia("(min-width: 400px)").matches) {

/\* the view port is at least 400 pixels wide \*/

} else {

/\* the view port is less than 400 pixels wide \*/

}

根据Unicode 值，然后返回一个字符串 :

String.fromCharCode(72,69)

onMouseOver={Proxy => {

setAttr(

Proxy.currentTarget,

'/static/common/images/integration/equity/2\_8\_on.gif'

);

}}

function setAttr(selector, src = '') {

$(selector).attr('src', src);

}

数组比较 ：

import deepEqual from 'deep-equal';

deepEqual()

js 取页面中所有的选中文字 ： window.getSelection().toString()

系统时间 ： <https://www.cnblogs.com/carekee/articles/1678041.html>

**========================================================**

React

import { AppContainer } from "react-hot-loader"; //热更新

import { PersistGate } from "redux-persist/lib/integration/react"; //持久化

componentWillMount 在渲染前调用,在客户端也在服务端。

componentDidMount : 在第一次渲染后调用，只在客户端。之后组件已经生成了对应的DOM结构，可以通过this.getDOMNode()来进行访问。 如果你想和其他JavaScript框架一起使用，可以在这个方法中调用setTimeout, setInterval或者发送AJAX请求等操作(防止异部操作阻塞UI)。

当前props和上一个props做比较 ：

componentDidUpdate(prevProps, prevState) {

const { selectedDetailAccount: cur } = this.props

const { selectedDetailAccount: pre } = prevProps

if (cur && pre && cur.accNumber != pre.accNumber) {

this.handleRefine()

}

}

**========================================================**

JSX

**========================================================**

Redux

redux 数据流 :

Action --- Reducers方法 --- 更新 Store --- 更新 UI --- Action

action(定义接口) --- reducer(将response封装到state) --- export default connect(我想要的数据mapStateToProps，我想要的方法mapDispatchToProps)

与后台交互 ：

action 文件夹中 定于ACTION

--- services文件夹中 定义ajax路径

--- reducers文件夹中 定义ajax成功/失败返回的值放入state中，并且 可以设置initState初始化。

.jsx 调 actionFun

action/.js 调 servicesFun

services/ .js 调 INDXACTION

componentWillMount() {

this.props.dispatch(actionRequest(MYAGENDA\_INIT)); // initialize

}

componentDidMount() {

initJqueryJs();

}

componentDidUpdate() {

initJqueryJs();

}

**========================================================**

npm修改

例子 ： npm config set proxy http://http.proxy.fmr.com:8000/

npm config set https-proxy = "http://http.proxy.fmr.com:8000/"

npm config set proxy = "http://http.proxy.fmr.com:8000/"

npm config set registry = "https://ftl.uk.fid-intl.com:8443/repository/FIL-NPM"

**========================================================**

VSCode

vscode merrge 的时候

列子 :

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

保留上边( 服务器的更改 )的代码 | 保留下面( 自己的更改 )的代码 | 都保留 |

<<<<<<< Updated upstream

import calendar from './calendar';

=======

import myAgenda from './myAgenda'

>>>>>>> Stashed changes

禁止保存的时候自动格式化 ： "editor.formatOnType": false,和"editor.formatOnSave": false

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

VSCode Shortcut keys

打开错误面板 ：Ctrl+Shift+M

快速跳转到错误位置 ： F8 or Shift+F8

更改语言模式 : Ctrl+K M

快速打开文件 ： Ctrl+P

命令板 ： Ctrl+Shift+P

更改主题 ：Ctrl+K Ctrl+T

实现滑轮 字体大小："editor.mouseWheelZoom": true

ctrl+shift+p

**========================================================**

Git

拉代码 ：

git clone

git status

git branch -r

git checkout 26(target\_branch)

git pull origin 26(sourc\_branch)

git checkout 3332(target\_branch)

git merge 26(sourc\_branch)

编写代码 ：

https://sdi.ap.fid-intl.com/jira/browse/HKWEBX-???? --- 创建分支(Create branch)

git fetch --all

git checkout 新建的分支

提交：

git status

git add \* --- 添加到缓存区

git commit -m "注释"

git push origin target\_branch

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

git remote -v --- 查看远程git地址

git branch -a --- 查看远程分支地址 --- （红色：远程分支列表） --- （绿色：当前项目所在分支）

git branch --- 查看本地分支地址

git branch -d xxxxx --- 删除本地分支

npm cache clean -f --- 清除本地缓存

git checkout -- 不提交的文件

查看对包的控制权限，可以使用npm access

查看npm有关的命令，可以使用npm h

添加用户相关信息，可以使用npm add-user

查看api帮助文档，使用命令npm apihelp

查看作者或创建人，使用npm author

查看npm安装的可执行文件路径，使用npm bin