@pyplane_code

Javascript window object

JS

part 1

@pyplane_code

01

Introduction

In this tutorial we will explore some of the most popular properties of the window object, which is a representation of an open window in a browser

contents of part 1:

- location
- history
- document
- navigator
- screen
- remarks

@pyplane_code

02

location

returns a location object which contains information about the current url

```
// reload the page after 2 seconds
setTimeout(()=> {
  window.location.reload()
}, 2000)
```

```
▼Location i
  ▶ancestorOrigins: DOMStringList {length: 0}
  ▶assign: ƒ assign()
   hash: ""
   host: "localhost:3000"
   hostname: "localhost"
   href: "http://localhost:3000/"
   origin: "http://localhost:3000"
   pathname: "/"
   port: "3000"
   protocol: "http:"
  ▶reload: ƒ reload()
  ▶replace: ƒ replace()
   search: ""
  ▶toString: ƒ toString()
  ▶valueOf: ƒ valueOf()
   Symbol(Symbol.toPrimitive): undefined
  ▶[[Prototype]]: Location
```

@pyplane_code

03

history

returns an history object which contains the pages visited by a user

```
// go back after 2 seconds
setTimeout(()=> {
  window.history.go(-1)
}, 2000)

// or
setTimeout(()=> {
  window.history.back()
}, 2000)
```

```
History {length: 1, scrollRestoration: 'auto', state: null}
  length: 1
  scrollRestoration: "auto"
  state: null
  [[Prototype]]: History
    back: ƒ back()
    forward: ƒ forward()
    go: ƒ go()
    length: (...)
    pushState: ƒ pushState()
    replaceState: ƒ replaceState()
    scrollRestoration: (...)
    state: (...)
    constructor: ƒ History()
    Symbol(Symbol.toStringTag): "History"
    get length: ƒ length()
    get scrollRestoration: ƒ scrollRestoration()
    set scrollRestoration: ƒ scrollRestoration()
    get state: ƒ state()
    [[Prototype]]: Object
```

document

returns the document object as an html document loaded into a web browser

```
// get the title of the page
console.log(window.document.title)
// <title>my app</title> -> my app

// change the body of the page
window.document.body.innerHTML = "hello world"
```

```
▼#document
   <!DOCTYPE html>
   <html lang="en">
   ▶<head>…</head>
     <body>hello world</body>
   </html>
```

@pyplane_code

05

navigator

returns and object containing information about the users browser

```
// preferred language of the user
console.log(window.navigator.language)

// get location coordinates
if (navigator.geolocation) {
  window.navigator.geolocation.getCurrentPosition(pos => {
    console.log(pos)
  })
} else {
  console.log("couldn't get the position")
}
```

@pyplane_code

06

screen

returns an object which contains the information about the users screen

```
console.log(window.screen.height)
// 900

console.log(window.screen.width)
// 1440
```

```
Screen {availWidth: 1440, availHeight: 900, width: 1440, height:
900, colorDepth: 30, …} i
  availHeight: 900
  availLeft: 0
  availTop: 0
  availWidth: 1440
  colorDepth: 30
  height: 900
  isExtended: false
  onchange: null
▶ orientation: ScreenOrientation {angle: 0, type: 'landscape-prima
  pixelDepth: 30
  width: 1440
▶ [[Prototype]]: Screen
```

@pyplane_code

07

remarks

when writing vanilla js code you can use shortcuts like in the examples below. If for example you are working on react app for location and history you may want to use hooks provided by react router dom

```
window.document.getElementById('root')
// or
document.getElementById('root')

window.location.reload()
// or
location.reload()

window.localStorage.setItem('name', 'Joe Doe')
// or
localStorage.setItem('name', 'Joe Doe')

window.history.back()
// or
history.back()


// in react
const history = useNavigate()
const location = useLocation()
```