Core JS

Env

VS code (Brackets extension pack, Snippet, TabOut, Open in External App)
Live preview
Chrome devtool

Input/Output

input

```
i=prompt(“enter a number”)
```

output

```
alert(message)
console.log(i)
document.write(i)
h1.innerText=aValue
```

Variable

```
let n=100;
var n=100;
```

Objects

```
// create, read, update, delete
person={
   name: 'Joe',
   age:20,
   hobbies: ['hiking','tennis'],
   getInfo: function() {
     return name+" with age"+age;
   }
}
// read, update, delete keys

// shallow copy:
  a={
    key1:'v1',
    key2:'v2',
    key3: {}
  }
  b={...a} //a.key3==b.key3

// deep copy
c=structuredClone(a);
// c.key3!=a.key3
```

== vs ===

function

```
function add(a,b) { return a+b}
// missing para no error
add(100)
// extra para ignored
add(100,200,300)

// auto invoke function
a=(function (a,b) {
    return a-b;
})(3,100);
```

Canvas

```
<canvas width=700 height=500>

canvas=document.querySelector('#canvas');
ctx = canvas.getContext(‘2d’);
```

Canvas size (optional)

element size vs display size

```
canvas.width
canvas.style.width
getComputedStyle(canvas).width
```

canvas resize:

```
canvas.width, canvas.height
```

canvas (rect, arc, line)

```
// rectangle
ctx.fillRect(x, y, w, h)
// flip rectangle
ctx.rect(x,y, -width, -height)
//
ctx.rect(x, y, w, h);
ctx.fillStyle=‘red’;
ctx.fill();

// rectangle (stroke)
ctx.rect(x, y, w, h)
ctx.strokeStyle=‘red’
ctx.lineWidth=10
ctx.stroke()

// arc
ctx.arc(x, y, r, 0, Math.PI*2)
// degree -> radian
45 deg = PI*/180*45 radian

// line
ctx.moveTo(x,y);
ctx.lineTo(x1,y1);
ctx.lineTo(x2,y2);
ctx.strokeStyle=‘red’;
ctx.closePath();
ctx.stroke();
ctx.lineCap=’round’ (optional)
ctx.lineJoin=’round’ (optional)

// begin to make a new path list
ctx.beginPath()
ctx.arc(x,y,r,0,Math.PI*2)
ctx.strokeStyle=‘blue’
ctx.stroke();
```

Callback function

a function passed into another function

Loop

for, for in, for of
while, do while

```
a.forEach(function(e, index) {})
```

game loop 1

```
i=setInterval(cb,1000)
clearInterval(i)

requestAnimationFrame(loop)

// 3 steps
1 clear canvas
2 draw
3 update
```

key events

keydown, keyup

```
e.code===‘ArrowLeft’
```

debugging

1 console.log
2 set breakpoint, step over
3 ‘debugger’ keyword

mouse events

mouseover, mouseleave, mousemove
click, dblclick, wheel

collision

hit a rect with a circle

array

```
a=[1,2,3];
a=new Array(3).fill(0);
```

methods:

```
a.push(e), a.pop()
a.unshift(e), a.shift()
a.splice(startIndex, deleteCount, item1, ..., itemN)
```

canvas (text)

```
ctx.font=‘30px monospace'
ctx.fillText(text, x, y)

ctx.textAlign=‘center’ // left, right
ctx.textBaseline=‘middle’ // top, bottom

ctx.lineWidth=10;
ctx.strokeStyle=‘blue’
ctx.strokeText=‘Friend’
```

canvas (audio)

```
// create
a=document.querySelector(‘#audio’)
or
a=new Audio(‘miracle.mp3’)

// 1 properties
a.src, a.duration, a.volumn, a.currentTime

// 2 methods
a.play(), a.pause()

// 3 events
loadedmetadata, play, pause, volumechange, ended, timeupdate
```

canvas (image)

```
// create
img=new Image()

// properties
img.src=‘anim.png’
img.width, img.height
// events
img.addEventListener(‘load’, cb)

// draw image
ctx.drawImage(img, dx, dy)
// scale image
ctx.drawImage(img, dx, dy, dw, dh)
// draw part of image
ctx.drawImage(img, sx, sy, sw, sh, dx, dy, dw, dh)

// html ele -> js object
img=document.querySelector(‘#img’)
// js object -> page
img.style=’border: solid 2px red’
document.body.appendChild(img)

// canvas->img data->img->page
screenshot=canvas.toDataURL()
img.src=screenshot
document.body.appendChild(img)
```

gradient

```
//
g=ctx.createLinearGradient(x1,y1,x2,y2)
g.addColorStop(0, ‘white’)
g.addColorStop(1, ‘black’)
ctx.fillStyle=g

//
rg=ctx.createRadialGradient(x1,y1,r1,x2,y2,r2)
rg.addColorStop(0,’white’)
rg.addColorStop(1,’blue’)
ctx.fillStyle=g
```

Module

```
// module1.js
let a=100;
function b() {  return 200;}

export {a,b}

// module2.js
import {a} from ‘./module1.js’
let c=300;
function d() {  return 400;}
let e=a+500;

export {c,d,e};

// main.js
import {c,d,e} from “./module2.js”;
import {a as aa, b} from “./module1.js”;

// testModule.html
 <script src="main.js" type=“module"></script>
```

lambda

```
function a() {
  return 100;
}
( )=> {
  return 100;
}

// example
c.map((e)=>{
  return e*2
})
```

game loop 2 (interval)

```
let pre=null;
function loop() {
    requestAnimationFrame(loop);
    // set interval to 1 sec
    let curr=Date.now();
    if (curr-pre<1000) return;
    pre=curr;
    // do something
    console.log(‘hello’)
}
loop();
```

Closure

a function that is returned by another function

```
function a() {
 let price=100;
 function getPrice() {
   return price;
 }
 function setPrice(aPrice) {
   price=aPrice;
 }
 return {
   getPrice:getPrice,
   setPrice:setPrice
  }
}
```

function (named arguments)

```
function add(a=100,b=200) {
  console.log(a,b)
}

add()
add(10,20)
add(a=10)
add(b=20)
add(undefined,20)
```

random

```
Math.random() // [0, 1)
Math.floor(Math.random()*5) // [0,4]
```

data types

5 basic (object-like) types: number, string, boolean, null, undefined

others are all objects: object, array, function, regular expression (optional)

loop break/continue label

```
abc:
for (i=0; i<3; i++) {
   for (j=0; j<3; j++) {
      if (j==2) break abc;
    }
}
```

constructor

```
   function Person(name,age) {
     this.name=name;
     this.age=age;
     this.getAge=function() {
       return this.age;
     }
   }
// new Person(‘Fred’, 50) vs Person(‘Fred’, 50)

// ‘this’ in constructor
```

json

```
jsonObj=JSON.stringify(obj);
obj=JSON.parse(jsonObj);
```

Show how google doc use json

file load

```
 <input id='fileLoader' type=“file”>

// 1 get selected file
fileLoader.addEventListener('change', function (e) {
    console.log(e);
    // console.log(e.target.files);
    // console.log(fileLoader.files);

     // 2 read file
     let fileReader = new FileReader();
fileReader.readAsDataURL( fileLoader.files[0] );

fileReader.addEventListener('load', function (e) {
    // console.log(fileReader.result);

    let img=new Image();
    img.src=fileReader.result;
    img.addEventListener(‘load’, function(e){
        // 3 use file data
        ctx.drawImage(img,0,0);
      });
    });
});
```

Character jump

1 same speed
2 with gravity
3 jump on platform

spritesheet

Fire320x64x5.png

spritesheet_46rows_13cols_832x2944

1. anim all rows
2. only anim specified rows
3. move characters (standRight, runRight, standLeft, runLeft) by ArrowLeft, ArrowRight

transform the coordinate system of canvas

```
// translate, rotate, scale
ctx.translate(dx,dy)
ctx.rotate(radian)
ctx.scale(1,2)
ctx.scale(-1,2)

// skew
ctx.transform(1, 0, 0.5, 1, 0, 0)
/*
 a  Horizontal Scale
 b  Vertical Skew
 c  Horizontal Skew
 d  Vertical Scale
 e  Horizontal translate
 f  Vertical translate
*/

// rotate by transform function
ctx.transform(1, 0.5, -0.5, 1, 0, 0);

//
ctx.save() // save to a stack
ctx.restore() // pop from stack
ctx.resetTransform() // equal to ctx.setTransform(1,0,0,1,0,0);

// reset canvas (including clearing contents and reset transform)
canvas.width = canvas.width
```

async

https://github.com/latentflip/loupe

setTimeout(cb, 1000) // why at least 1 sec, not exact 1 sec

hoisting

before run code, JS first scan the code and move all declarations to the top of the current scope

```
// variale hoisting
a = 100;
var a; // declaration
```

```
// function expression hoisting
// b(); // b is not defined
var b=function() {
  console.log('hello');
};
b(); // no problem

// function declaration hoisting
c(); // no problem
function c() {
     // var a;
     a=100;
     var a;
}
```

‘this’ in event listener

event listener callback function creates its own ‘this’ pointing to e.target

event listener arrow function doesn’t create its own ‘this’

null vs undefined

undefined: programmer not provide a value

null: programmer provide a null value

Game projects

breakout
tetris

Optional

```
// transparency
ctx.globalAlpha=0.5
ctx.fillStyle = ‘rgba(0,255,0,0.5)';

// shadow
ctx.rect(x,y,w,h)
ctx.shadowOffsetX = 10;
ctx.shadowOffsetY = 10;
ctx.shadowBlur = 10;
ctx.shadowColor = ‘gray’;
ctx.fill()

// check type
typeof a
a instanceof b
```

(optional) Clip

// Use ctx.clip() to restrict drawing to a specific area on the canvas.

1. draw a path
2. ctx.clip() //use the path as clip path
3. draw any shape or path which will only show in the clip path

(optional) composite

(optional) Other applications

image processing
audio processing
video processing
drawing tool

(optional) More workflow

```
throw new Error(“message”)

try {
} catch (e) {
} finally {
}
```

Final assignment to get a certificate

breakout: more powerful balls, more stronger bricks

tetris: add sound, allow to review the next 5 tetrominos

Add 2 game projects to your personal webpage

Next step

phaser.js, or p5.js

5 skills

1. Drawing elements
2. Animating elements
3. Interacting with elements
4. Transform elements
5. Composite elements (optional)

Teens Programming