

JavaScript & Web

程式設計基本觀念

Week 3

課前(「課程複習」)

console.log()、變數、prompt()、
If - else if - else、+ - * / % =、
Object.property、object.method

課前('作業解答')

Console.log('坐好囉')

作業 1 – 期末成績計算

- 計算方式：兩次期中考各佔30%、期末考佔40%
- Input：50 40
- 計算過程：
$$50 * 0.3 + 40 * 0.3 + f * 0.4 = 60$$
$$f = (60 - 50 * 0.3 - 40 * 0.3) / 0.4 = 82.5$$
- Output：能及格的期末成績：82.5

作業 1 – 期末成績計算

- 計算方式：兩次期中考各佔30%、期末考佔40%
- Input：50 40
- 計算過程：
$$50 * 0.3 + 40 * 0.3 + f * 0.4 = 60$$
$$f = (60 - 50 * 0.3 - 40 * 0.3) / 0.4 = 82.5$$
- Output：能及格的期末成績：83 (~~82.5~~)

作業解答 – 期末成績計算

```
var m1 = prompt('第一次期末考成績');
var m2 = prompt('第二次末考成績');

var f = (60 - (m1 * 0.3 + m2 * 0.3)) / 0.4;
f = Math.ceil( f );

if(f > 100){
    console.log('快點去教務處拿二退單吧' );
}else if(f < 0){
    console.log('期末考在家睡覺吧！ 躺著過！ ');
}else{
    console.log('能及格的期末成績：' + f );
}
```

作業 2 – 閏年判斷

- 西元年份除以 4 不可整除，為平年。
- 西元年份除以 4 可整除，且除以 100 不可整除，為閏年。
- 西元年份除以 100 可整除，且除以 400 不可整除，為平年
- 西元年份除以 400 可整除，為閏年。
- 口訣：四年一閏，逢百不閏，逢四百又閏

作業 2 – 閏年判斷

Input

2015

2016

1900

2000

Output

2015 年不是閏年

2016 年是閏年

1900年不是閏年

2000年是閏年

作業解答 – 閏年判斷

整理後條件判斷：400整除 > 100 整除 > 4整除 > 其他年份

```
var year = prompt( '輸入西元年：' );
var result = '';
if( year % 400 == 0 ){
    result = '閏';
}else if( year % 100 == 0 ){
    result = '平';
}else if( year % 4 == 0 ){
    result = '閏';
}else{
    result = '平';
}
console.log( year + '年是' + result + '年' );
```

作業解答 – 閏年判斷

進階整理後條件判斷：如果(被400整除)或(被4整除並不被100整除)就是閏年

```
var year = prompt( '輸入西元年：' );
```

```
var result = '平';
```

```
if( year % 400 == 0 || year % 4 == 0 && year % 100 != 0 ) {  
    result = '閏';  
}
```

```
console.log( year + '年是' + result + '年' );
```

作業解答 – 閏年判斷

進階整理後條件判斷：如果(被400整除)或(被4整除並不被100整除)就是閏年

```
var year = prompt( '輸入西元年：' );
```

```
var result = '平';
```

```
if( (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0) ){  
    result = '閏';  
}
```

```
console.log( year + '年是' + result + '年' );
```

正課('第一部分')

Console.log('開始囉')

嘗試以下程式碼

```
1 console.log( Math.random() );  
2  
3 for(var i=0 ; i<100 ; i++){  
4     console.log(Math.random());  
5 };
```

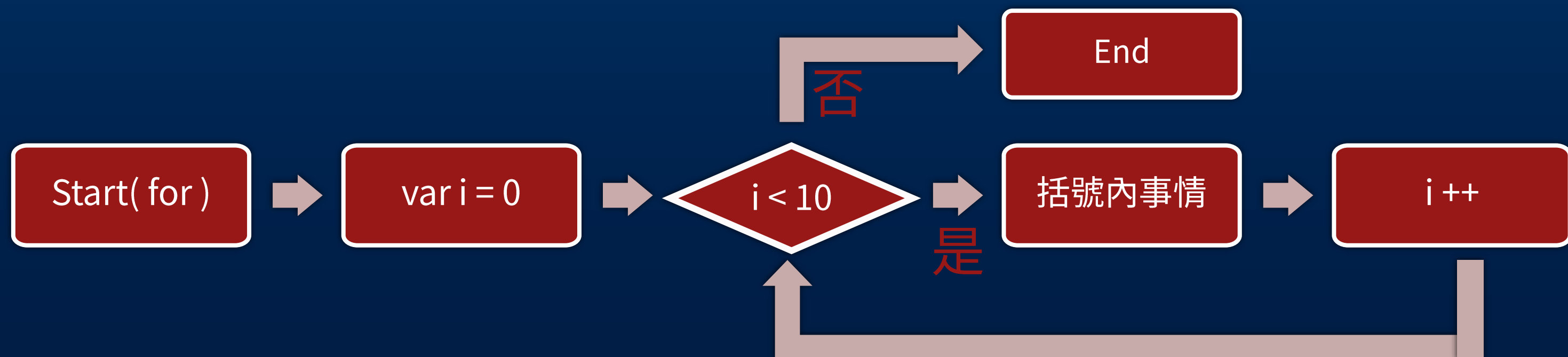
嘗試以下程式碼

```
1 console.log( Math.random() );  
2  
3 for(var i=0 ; i<100 ; i++){  
4     console.log(Math.random());  
5 };
```



For 迴圈流程圖

```
1 for(var i=0 ; i<10 ; i++){  
2   console.log(i);  
3 };
```



嘗試以下程式碼

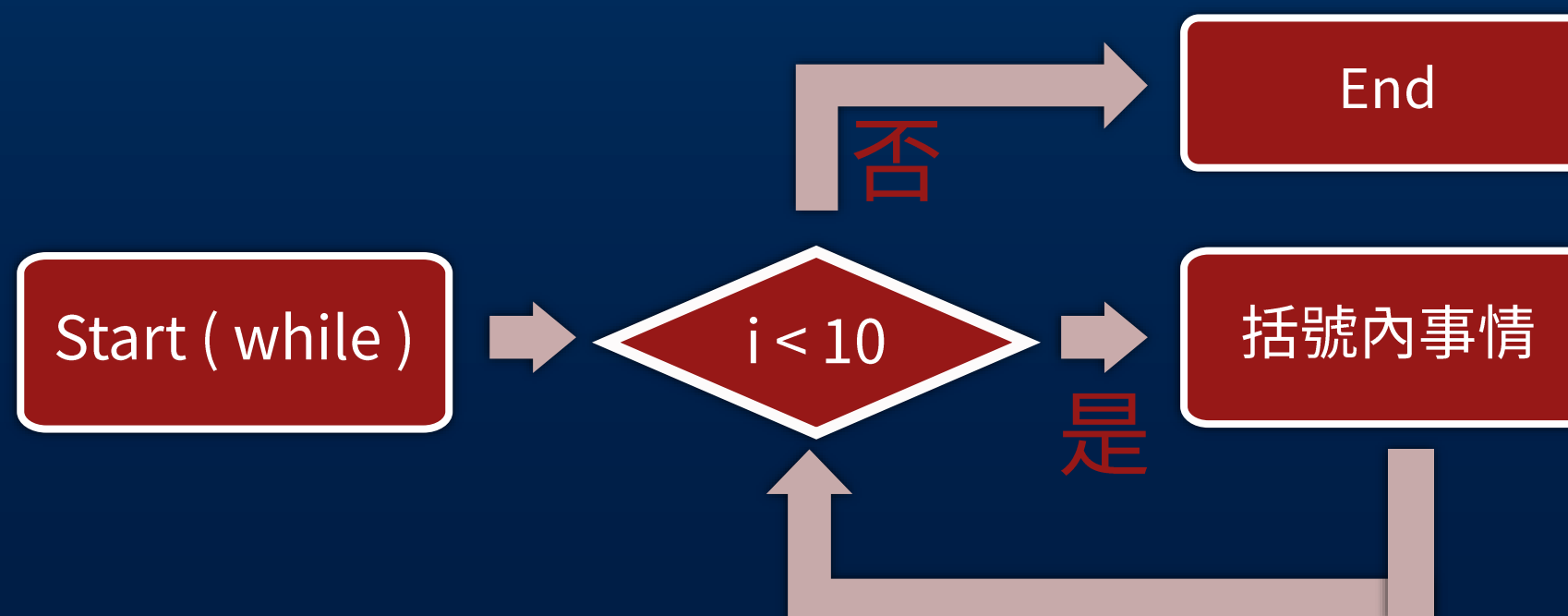
```
1 for(var i=1;i<=10;i++){  
2     if(i==4){  
3         continue;  
4     }  
5     if(i==8){  
6         break;  
7     }  
8     console.log(i);  
9 }
```


嘗試以下程式碼

```
1  var i =1;  
2  while( i <= 10){  
3      console.log(i);  
4      i++;  
5  }
```

while 迴圈流程圖

```
1  var i =1;  
2  while( i <= 10){  
3      console.log(i);  
4      i++;  
5  }
```



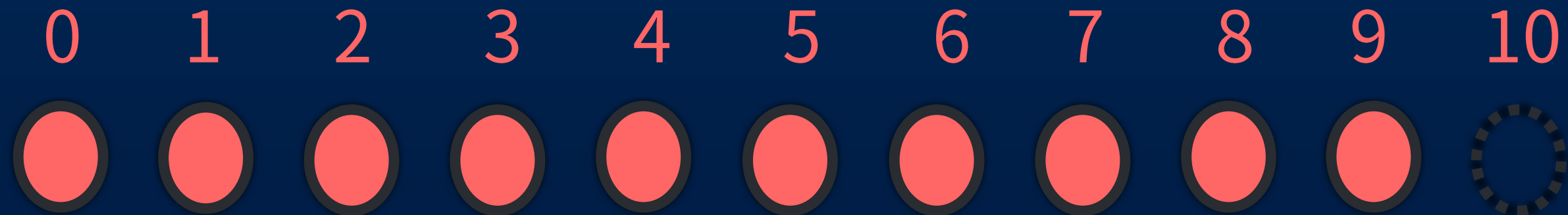
嘗試以下程式碼()

```
1 var n = Math.random()*10; // 範圍變成了???  
2 console.log( n );  
3 console.log( Math.floor(n) ); //與上行結果差異  
4  
5 for(var i=0 ; i<20 ; i++){  
6     console.log( Math.floor( Math.random()*10 ) );  
7 }
```



嘗試以下程式碼()

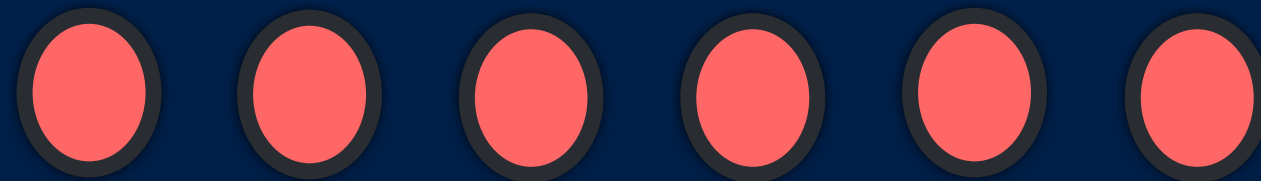
```
1 var n = Math.random()*10; // 範圍變成了???  
2 console.log( n );  
3 console.log( Math.floor(n) ); //與上行結果差異  
4  
5 for(var i=0 ; i<20 ; i++){  
6     console.log( Math.floor( Math.random()*10 ) );  
7 }
```



嘗試以下程式碼()

```
1 // 測試並思考看看範圍在哪
2 var n = Math.floor(Math.random()*6)+1;
3
4 //測試看看有無在在你想的指定範圍內
5 for(var i =0 ; i<6 ;i++){
6     console.log(Math.floor(Math.random()*6)+1);
7 }
```

0 1 2 3 4 5



嘗試以下程式碼()

```
1 // 測試並思考看看範圍在哪
2 var n = Math.floor(Math.random()*6)+1;
3
4 //測試看看有無在你想的指定範圍內
5 for(var i =0 ; i<6 ;i++){
6     console.log(Math.floor(Math.random()*6)+1);
7 }
```

1 2 3 4 5 6



設定亂數的程式

```
1 //做出指定在 Min 到 Max 範圍內取亂數整數的程式
2 var Min = 1;
3 var Max = 6;
4
5 //亂數區間個數
6 var Range = Max - Min + 1;
7 //取出區間亂數位置
8 var Rand = Math.Random()*Range;
9 //取整數並平移到指定範圍內
10 var Random_number = Math.floor(Rand)+Min;
```

設定亂數的程式

```
1 //取整數亂數程式
2 var Min = 1;
3 var Max = 6;
4 Math.floor(Math.random()* (Max - Min + 1)) +Min
5
6 //(Max - Min + 1) 為計算出取樣的個數
7 //floor(Math.random()* (Max - Min + 1))
8 //上面這行程式碼取出 0 ~ (Max - Min) 的亂數整數
9 // 0 ~ (Max - Min) 平移Min 變成 Min ~ Max
```


鐘聲(' 叮咚~叮咚~')

Console.log(' 上上廁所、多認識一下同學')

正課('第二部分')

Console.log('上半部還有甚麼問題嗎')

嘗試以下程式碼()

```
1 function HelloWorld1(){
2     alert('HelloWorld');
3 }
4 //HelloWorld1();
5 -----
6 function HelloWorld2( name ){
7     alert('Hello ' + name );
8 }
9 //HelloWorld2('JavaScript');
```

嘗試以下程式碼()

```
1  var my_alert = alert;
2  //my_alert = alert('XD');
3
4  -----
5  function HelloWorld( name ){
6      alert('Hello ' + name );
7  }
8
9  New_HelloWorld = HelloWorld;
10 //New_HelloWorld('JavaScript');
```

嘗試以下程式碼()

```
1 function my_add(min,max){  
2     var sum = 0;  
3     for(var i =min;i<=max;i++){  
4         sum += i;  
5     }  
6     return sum    ;//回傳結果  
7 }  
8  
9 // my_add(1,10)
```

函式 (function)

函式：一種包裝，減少複製貼上、方便開發

迴圈 由（if + 變數 + 回到指定位置 + 運算）組成
方便開發：random()、floor()、for()、while()...

特性：函式內變數與函式外無關、變數是幫函式取名字

如何寫：

```
1 function func_name( argument_name ){  
2     //statement  
3     return result ;//回傳結果  
4 }
```

函式 指定範圍整數亂數 ()

```
1 //回顧前面部分
2 var Max = 10;
3 var Min = 6;
4 Math.floor(Math.random()* (Max-Min+1) )+Min
5 -----
6
7 function _____(Min,Max){
8     //statement
9     //return
10 }
```

函式 指定範圍整數亂數 ()

```
1 //回顧
2 var Max = 10;
3 var Min = 6;
4 Math.floor(Math.random()* (Max-Min+1) )+Min
5 -----
6 function GetRandom(Min,Max){
7     var Range = Max - Min + 1;
8     var Rand = Math.random()*Range;
9     var number = Math.floor(Rand)+Min;
10    return number;
11 }
12 // GetRandom(1,6);
```


函式 指定範圍整數亂數 ()

```
1 //簡化
2 function GetRandom(Min,Max){
3     var Range = Max - Min + 1;
4     var Rand = Math.random()*Range;
5     var number = Math.floor(Rand)+Min;
6     return number;
7 }
8 -----
9 function GetRandom(Min,Max){
10     return Math.floor(Math.random()* (Max-Min+1) )+Min;
11 }
```

綜合練習 – 猜數字遊戲

- 由 電腦 出題目
- 由 player 猜答案
- 在猜數字前 電腦 會說出目前範圍
- player 能猜一個在範圍內的數字
- 如果 player 猜錯，電腦 會更新目前範圍
- 重複以上三點直到猜對為止

綜合練習 – 猜數字遊戲

數字介於 1 到 100 之間

☐ 防止此網頁產生其他對話方塊。

確定 取消

0



100



綜合練習 – 猜數字遊戲

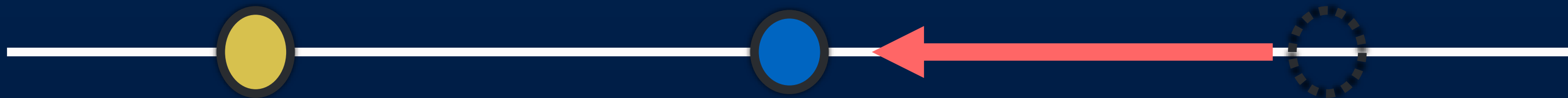
數字介於 1 到 100 之間

猜的數字太大囉

數字介於 1 到 50 之間

0

50

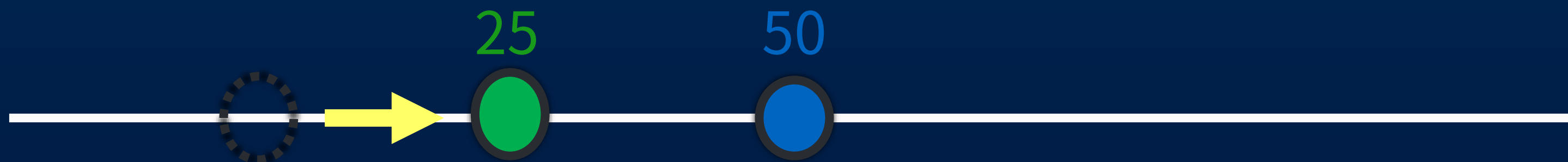


綜合練習 – 猜數字遊戲

數字介於 1 到 50 之間

猜的數字太小囉

數字介於 25 到 50 之間

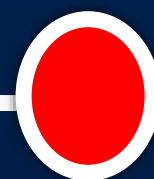


綜合練習 – 猜數字遊戲

數字介於 25 到 50 之間

你猜對囉

37



提示 – 猜數字遊戲

- 題目製造(); //範圍為 Min 到 Max 之間
- 輸出可猜測範圍(); //輸出目前Min 到 Max範圍
+
- 輸入猜測數字(); // prompt();
- 判斷猜測數字結果();
 - if(猜對) { 恭喜猜中 }
 - if(猜錯) { 說明猜測狀況、變更猜測範圍 }

提示 – 猜數字遊戲

```
1 //記得引入GetRandom(Min,Max)
2 var Min=1; var Max=100;
3 var answer = GetRandom( _____ , _____ );
4 var guess = 0;
5
6 for(var i=10 ; i<1 ; i--){
7     guess = prompt('輸入 ' + Min + ' ~ ' + Max + ' 間的數字');
8     if(_____){
9         alert('恭喜你答對了!');
10        _____;
11    }else if(_____){
12        alert('猜的數字太大囉');
13        ____ = _____;
14    }else{
15        alert('猜的數字太小囉');
16        ____ = _____;
17    }
18 }
```


解答 – 猜數字遊戲

```
1 //記得引入GetRandom(Min,Max)
2 var Min=1; var Max=100;
3 var answer = GetRandom( Min+1 , Max-1 );
4 var guess = 0;
5
6 for(var i=10 ; i<1 ; i--){
7     guess = prompt('輸入 '+ Min + ' ~ ' + Max + '間的數字');
8     if(guess == answer){
9         alert('恭喜你答對了!');
10        _____;
11    }else if(guess > answer){
12        alert('猜的數字太大囉');
13        Max = guess;
14    }else{
15        alert('猜的數字太小囉');
16        Min = guess;
17    }
18 }
```

報告('離開前的一句話')

`Console.log('老師今天怎麼沒回家作業')`

作業 – 因數分解

input

36

output

1 2 3 4 6 9 12 18 36

作業- 剪刀石頭布

input

paper

output

電腦出"Stone"

你贏了!!

input

paper

output

電腦出"Scissors"

你輸了!!

挑戰題 – 質因數分解

input

120

output

2 2 2 3 5

DeBug – 猜數字遊戲

數字介於 1 到 100 之間

☐ 防止此網頁產生其他對話方塊。

數字介於 1 到 100 之間

-100

☐ 防止此網頁產生其他對話方塊。

確定

取消

數字介於 1 到 100 之間

9999

☐ 防止此網頁產生其他對話方塊。

確定

取消

0



100



研究('新玩具')

Console.log('什麼是遞迴')

思考以下程式碼()

```
1 //階乘 想想程式怎麼達成的
2 function factorial(n){
3     if(n > 1){
4         return n * factorial(n-1)
5     }else{
6         return 1;
7     }
8 }
```


遞迴解說 - 階程

```
1 //階乘  $n! = n * (n-1)!$ 
2 function factorial(n){
3     if(n > 1){
4         return n * factorial(n-1)
5     }else{
6         return 1;
7     }
8 }
9 //問題簡化  $n! = n * (n-1)$ 
10 // $n! = n * (n-1) * (n-2) * \dots 1$ 
11 //終止條件式如果n為1時候就不在往前找
```

思考以下程式碼()

```
1 //fibonacci 數列
2 // 1、1、2、3、5、8、13、21、34.....
3 function fibonacci(n){
4     if(n > 2){
5         return fibonacci(n-1) + fibonacci(n-2)
6     }else{
7         return 1;
8     }
9 }
```

遞迴解說 - fibonacci

```
1 //fibonacci 數列
2 // 1、1、2、3、5、8、13、21、34.....
3 function fibonacci(n){
4     if(n > 2){
5         return fibonacci(n-1) + fibonacci(n-2)
6     }else{
7         return 1;
8     }
9 }
10 //第n項 = 第(n-1)項 + 第(n-2)項
11 //終止條件：如果是第1或2項不需要再往前找項
```

研究('活力補充飲')

Console.log('上課額外例子')

if(剛剛程式寫的是對的嗎)?

```
1 //測試機率
2 Min = 1; Max = 6;
3 var n = new Array(Max-Min+1); // var n = [0,0,0,0,0];
4 for(var i=0;i<n.length;i++){
5     n[i]=0;
6 }
7
8 repeat_n = 10000; //指定測試次數
9 for(var i=0 ; i<repeat_n ; i++){
10     number = Math.floor(Math.random()* (Max-Min+1) )+Min ;
11     n[ number - Min ] ++;
12 }
13
14
15 for(var i=0 ; i< n.length ; i++){
16     console.log('數字' + (i+Min) + '總共出現' + n[i] +
17         '次 機率約為 ' + (n[i]/repeat_n)*100 + '%');
18 }
```

嘗試以下程式碼()

```
1 function test_var( number ){
2     console.log( '[函式內]加減前 number = ' + number );
3     number++;
4     console.log( '[函式內]加減後 number = ' + number );
5 }
6
7 var number = 10;
8 console.log( '[函式外]執行前 number = ' + number );
9 test_var( number );
10 console.log( '[函式外]執行後 number = ' + number );
```