



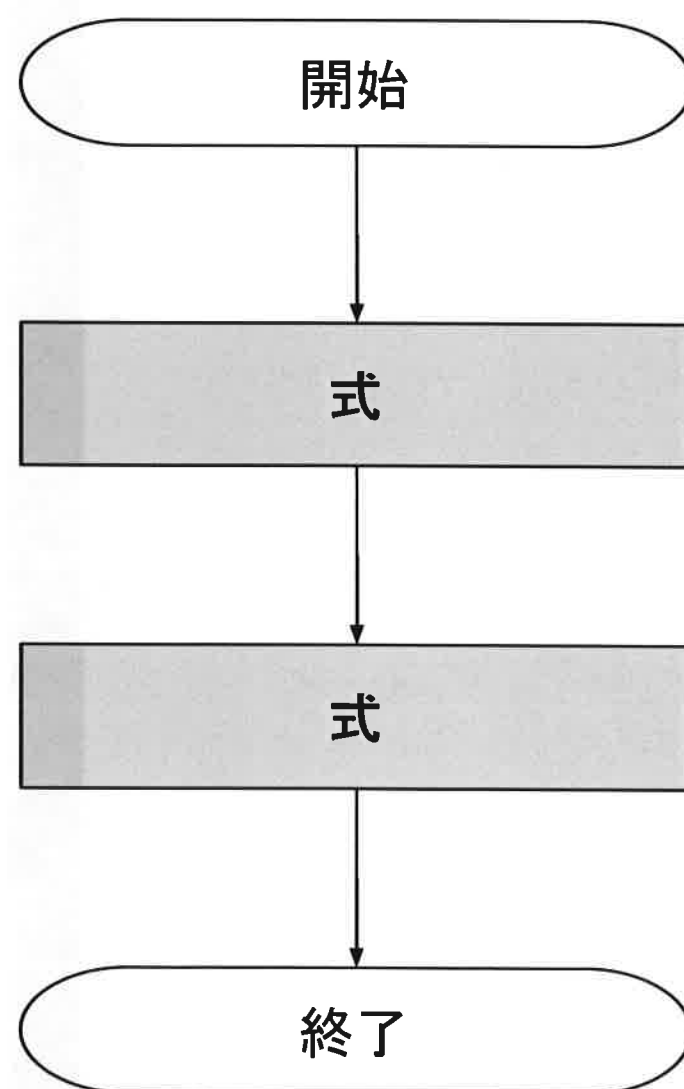
DAY02 ワークシート

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

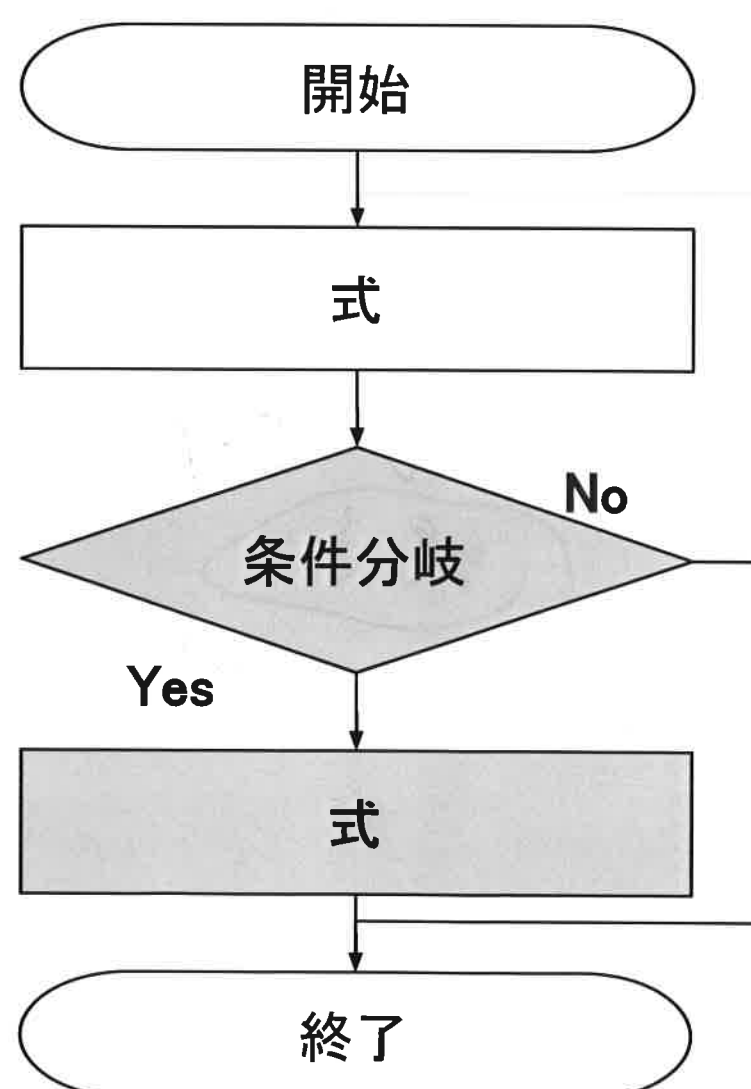
<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

フローチャートの書き方は、まず「順序実行、条件分岐、反復（繰り返し）」をおさえておけば大丈夫。

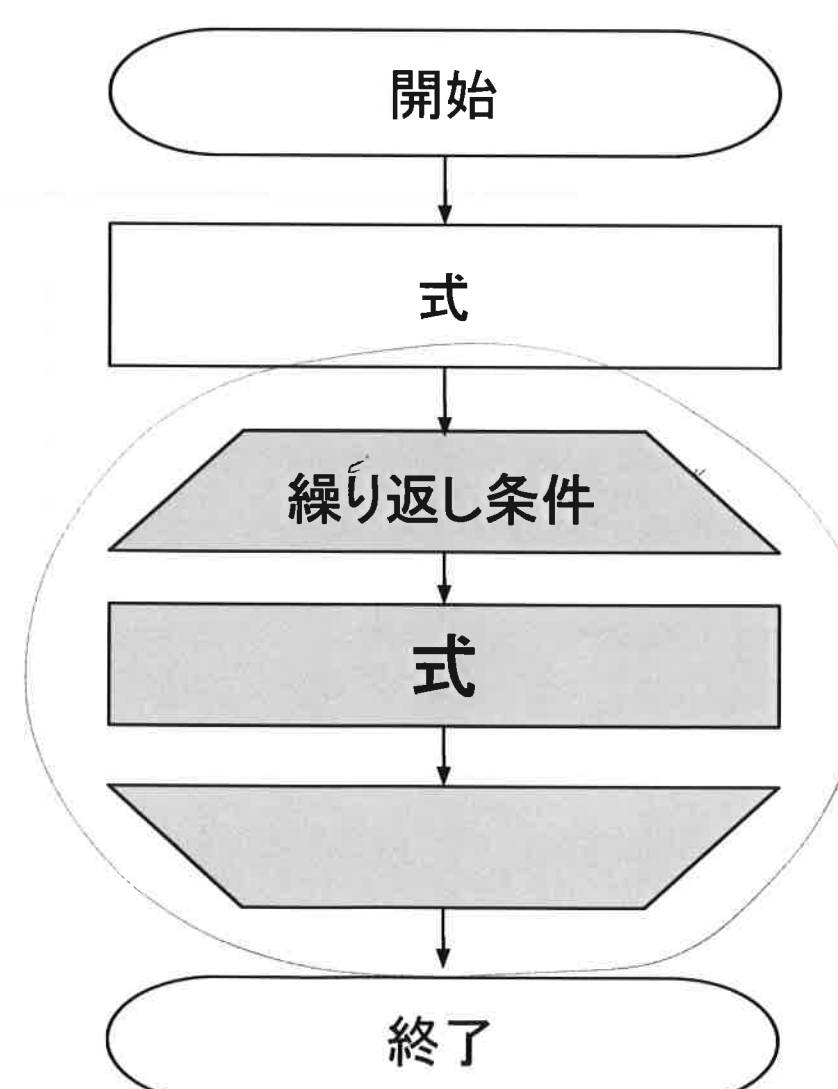
順序実行



条件分岐



反復（繰り返し）





DAY02 ワークシート

2月 期氏名: 八島友樹

JavaScriptシリーズ課題4【合否判定プログラム】

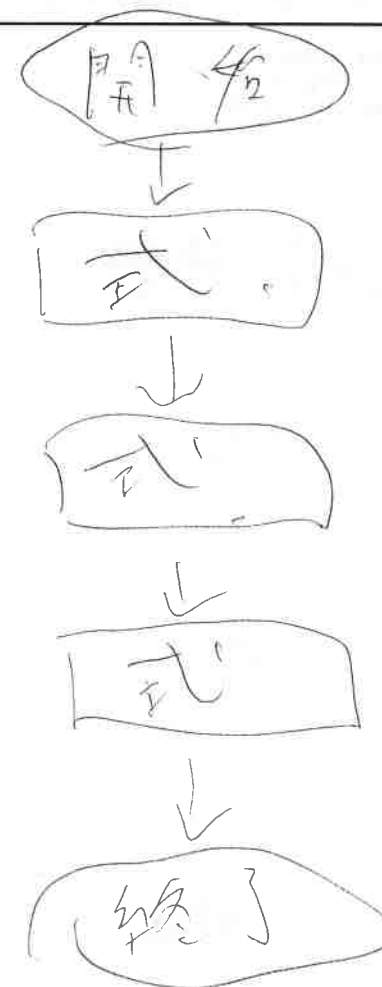
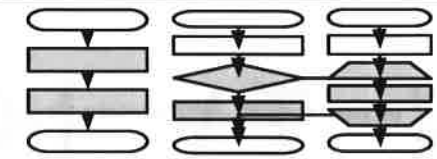
※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

①コードを書き出す

```
function judgement (points) {  
  let achievement = getachievement (points);  
  let pass_or_failure = get_pass_or_failure (points);  
  return `あなたの成績は${achievement}です。  
  ${pass_or_failure}です!`;  
}
```

②フローチャートをイメージする





DAY02 ワークシート

2月期氏名: 八島 友理

JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

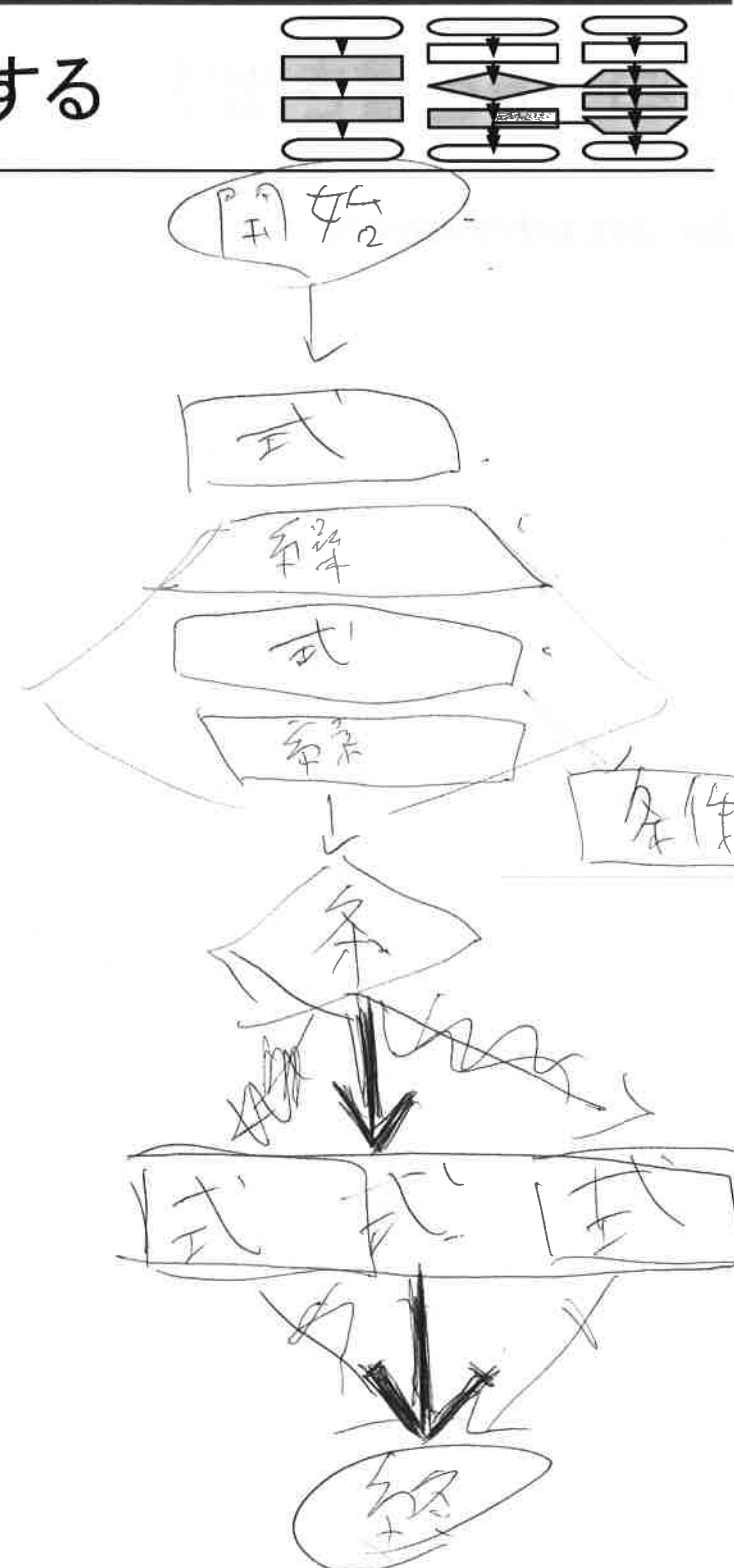
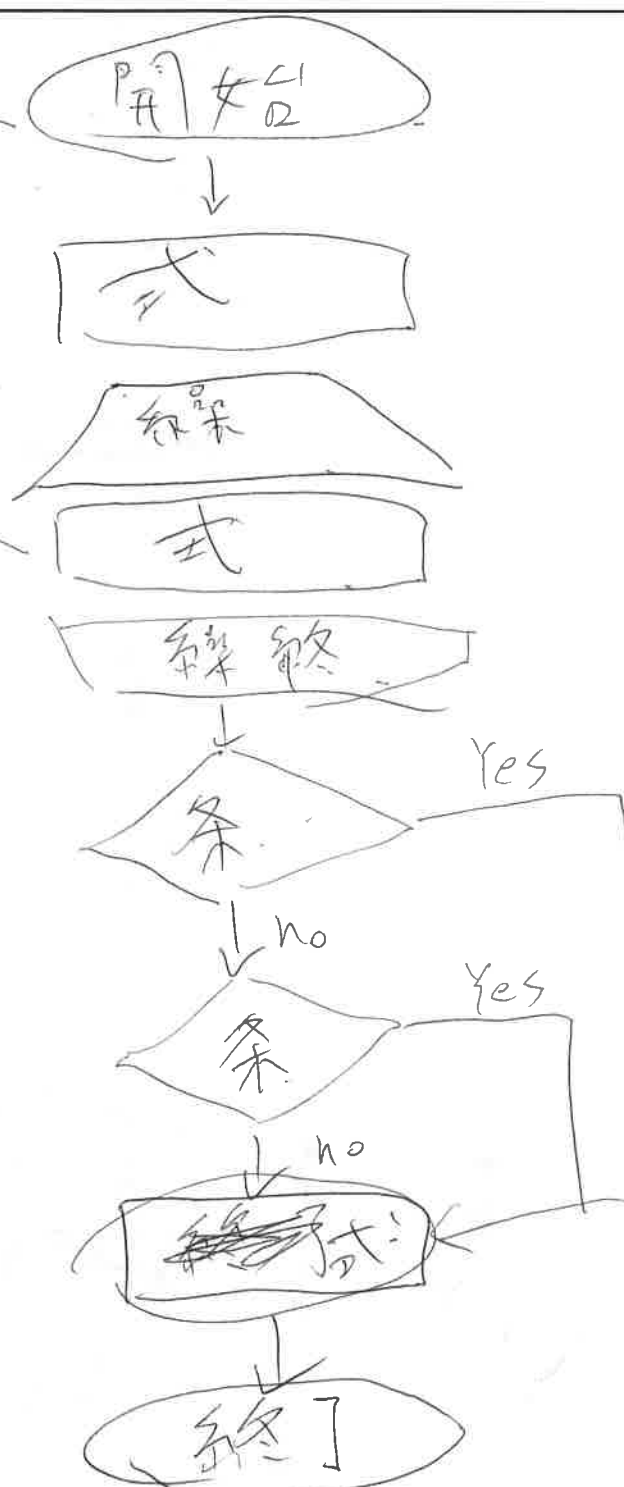
https://diver.diveintocode.jp/textbooks/329_newJudgement.js

①コードを書き出す

```
function get_pass_or_failure (points) {  
  let judge = "合格";  
  for (let i=0; i<number; i++) {  
    if (points[i]<60){  
      judge = "不合格";  
    }  
  }  
  return judge;  
}
```

break? 実行

②フローチャートをイメージする





DAY02 ワークシート

2月 期氏名: 八島 友樹

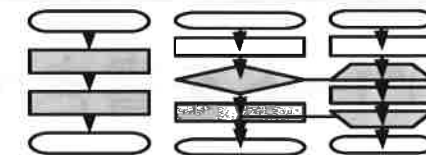
JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

https://diver.diveintocode.jp/textbooks/329_newJudgement.js

①コードを書き出す

②フローチャートをイメージする



```
function get_achievement (points){
```

```
  let sum = 0;
  for (let i = 0; i < number; i++) {
    sum = sum + points[i];
```

```
  }
  if (sum >= 80 * number) {
    return "A";
```

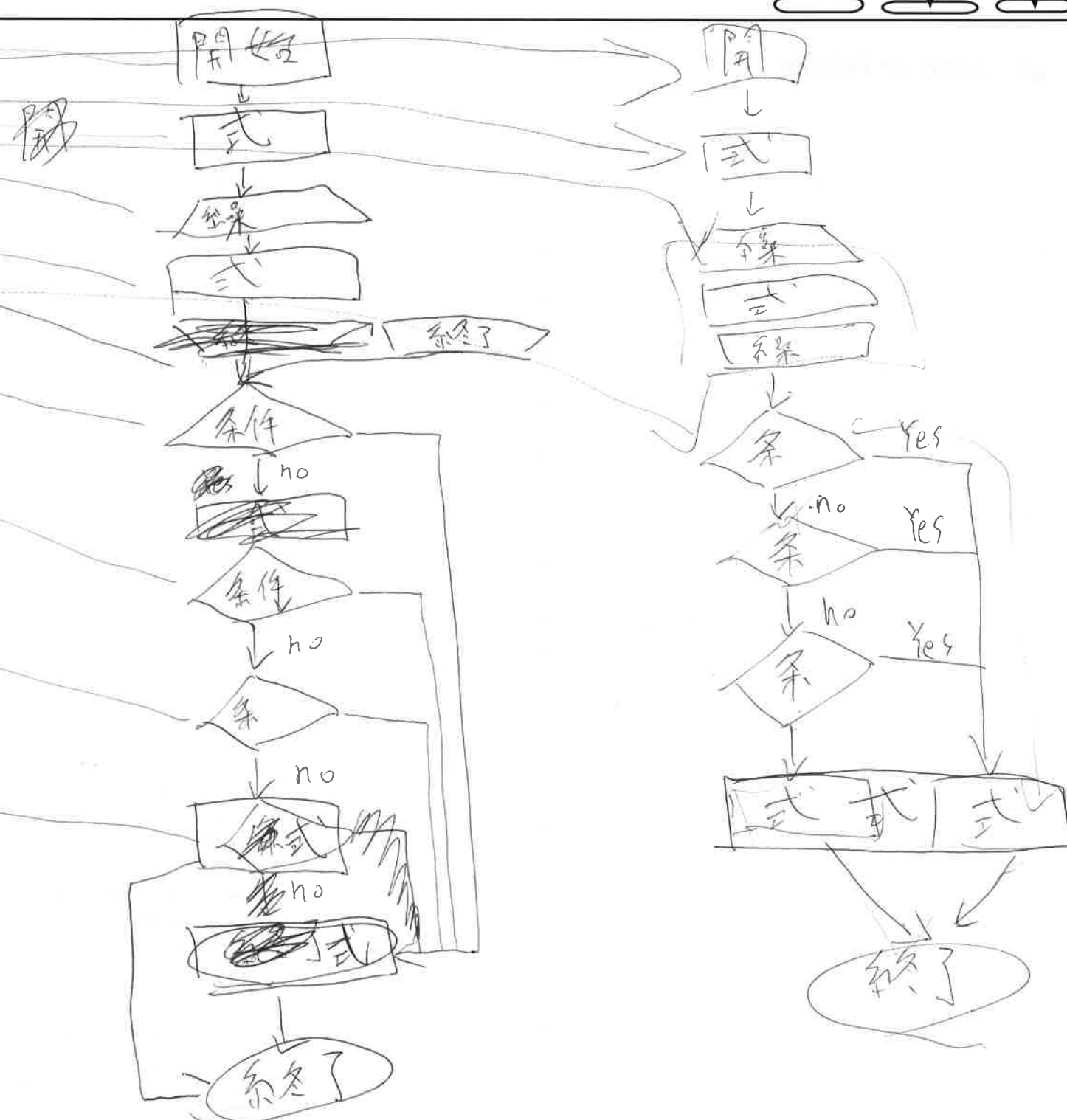
```
  } else if (sum >= 60 * number) {
    return "B";
```

```
  } else if (sum >= 40 * number) {
    return "C";
```

```
  } else {
    return "D";
```

```
  }
```

```
}
```





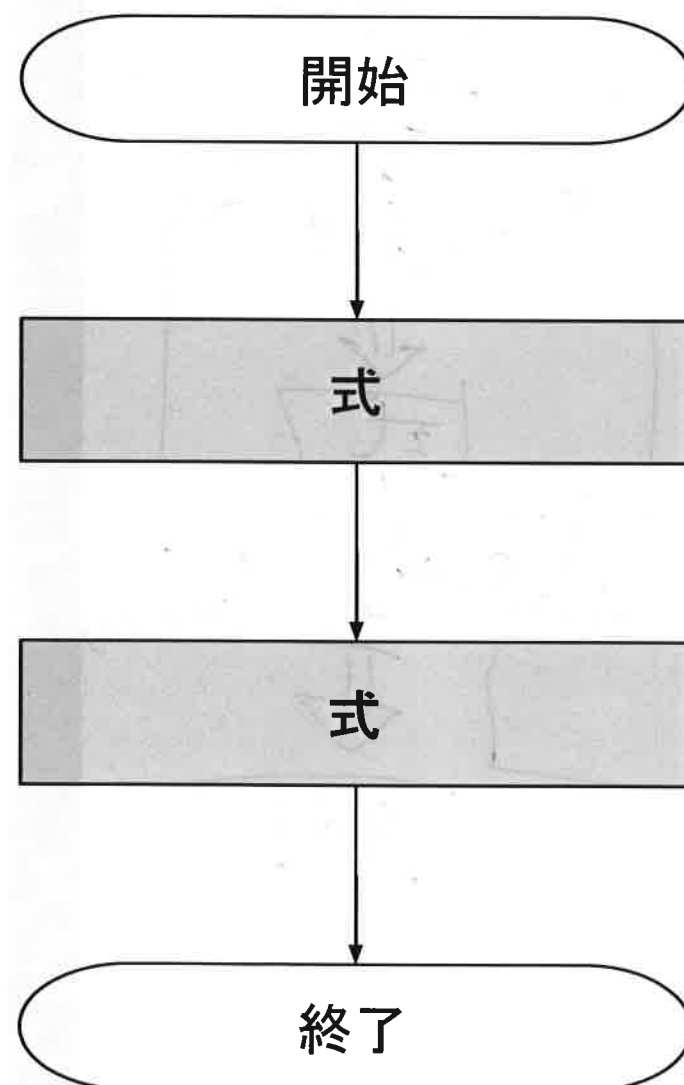
DAY02 ワークシート

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

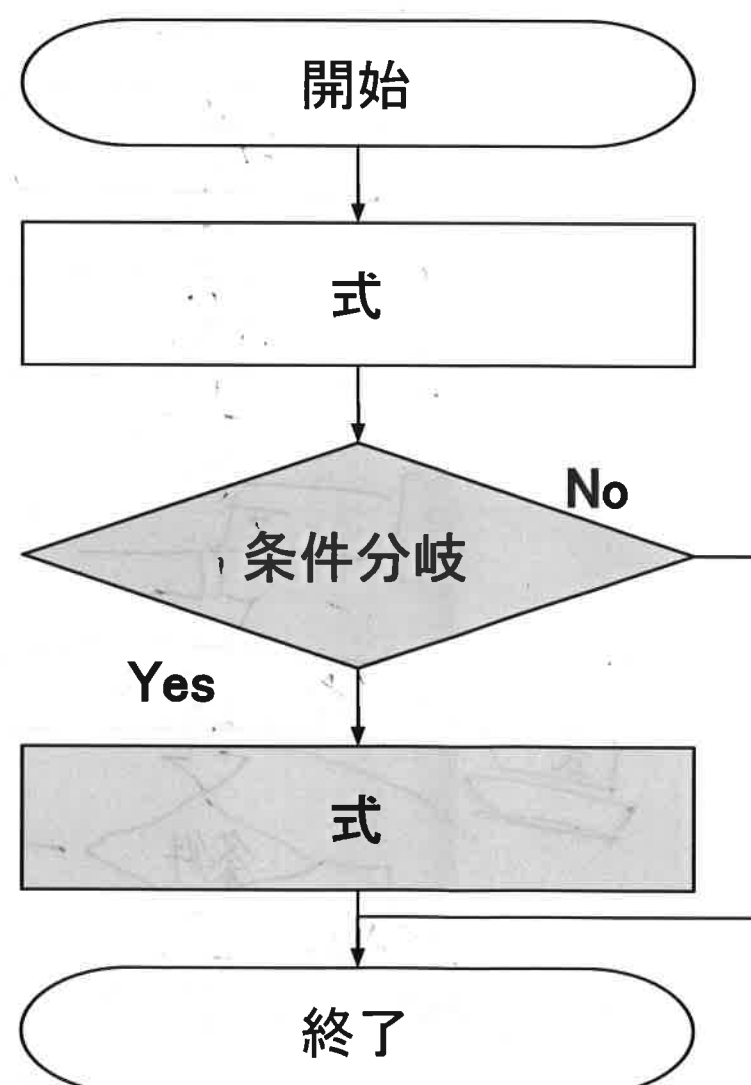
https://diver.diveintocode.jp/textbooks/329_newJudgement.js

フローチャートの書き方は、まず「順序実行、条件分岐、反復(繰り返し)」をおさえておけば大丈夫。

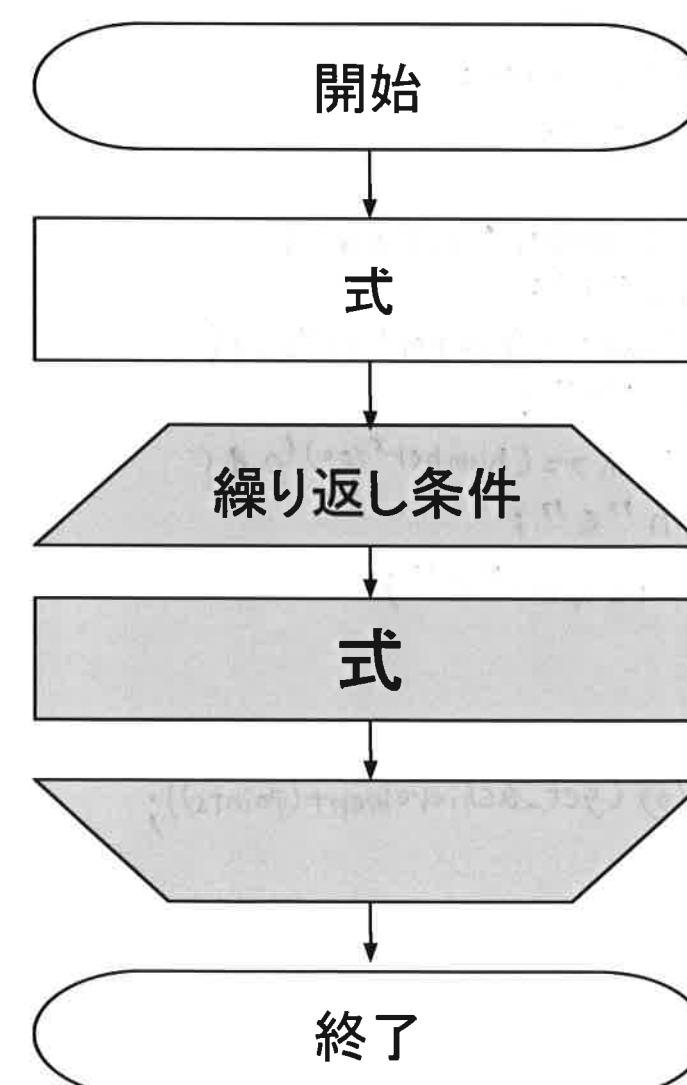
順序実行



条件分岐



反復(繰り返し)





DAY02 ワークシート

2月 期氏名: 安井 一

JavaScriptシリーズ課題4【合否判定プログラム】

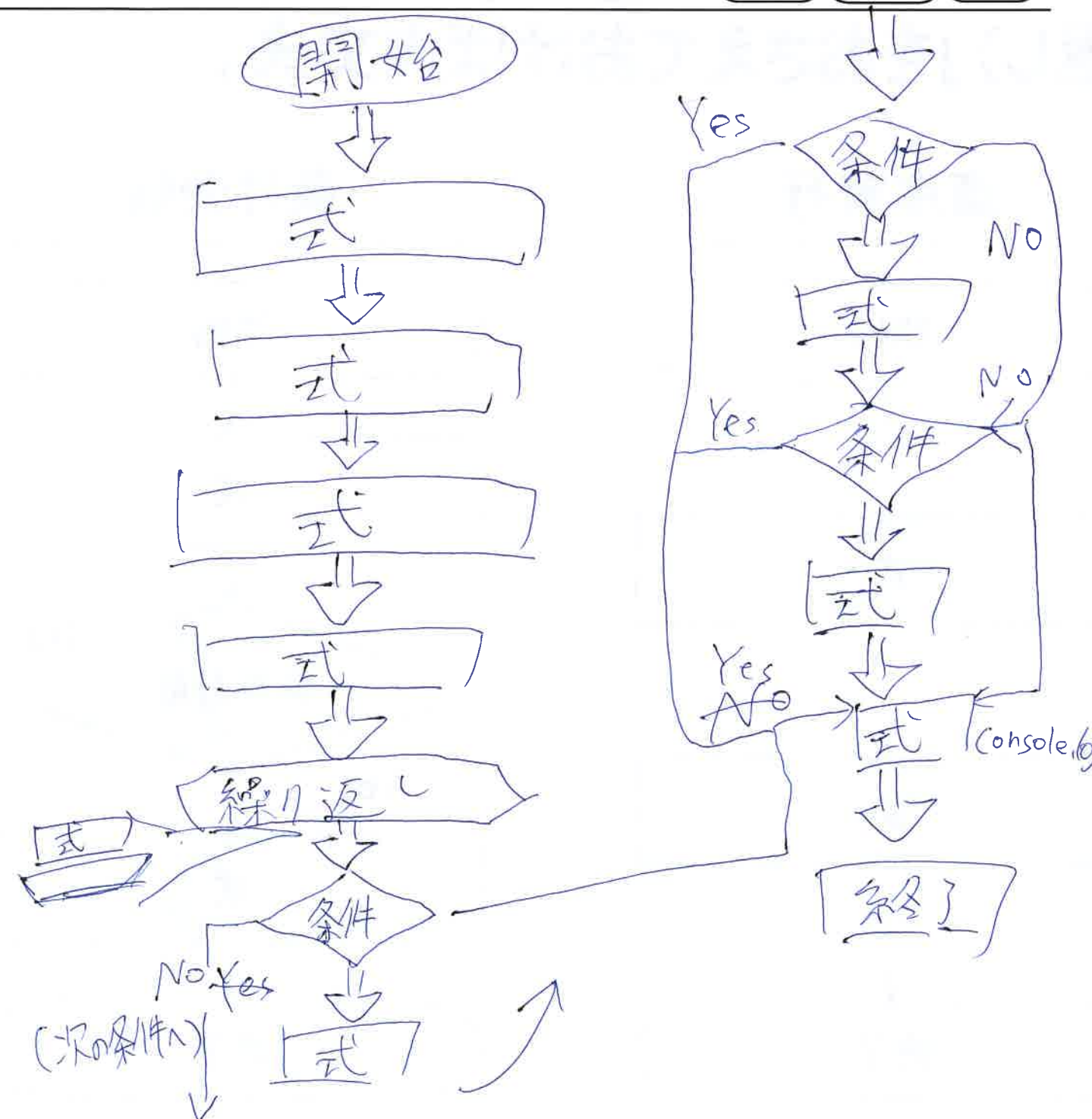
※授業後に回収します

https://diver.diveintocode.jp/textbooks/329_newJudgement.js

①コードを書き出す

```
function judgementget_achievement
let points = [60, 100, 80, 80, 70, 80];
let number = points.length;
function get_achievement(points) {
  let sum = 0;
  for(let i=0; i<number; i++){
    sum += points[i];
  }
  if(sum >= (number*100)*0.8) {
    return "A";
  } else if(sum >= (number*100)*0.6) {
    return "B";
  } else if(sum >= (number*100)*0.4) {
    return "C";
  } else {
    return "D";
  }
}
console.log(get_achievement(points));
```

②フローチャートをイメージする





DAY02 ワークシート

2月 期氏名: 安井

JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329/newJudgement.js>

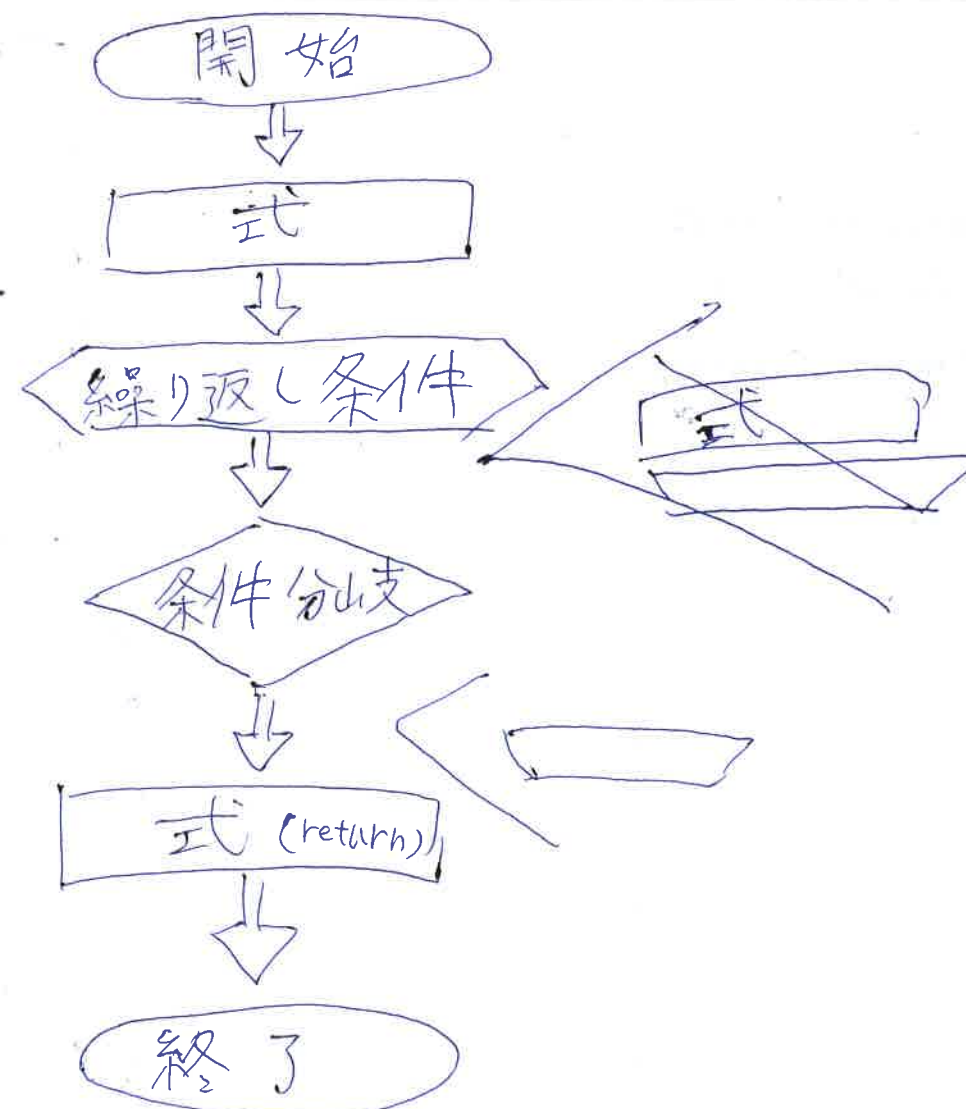
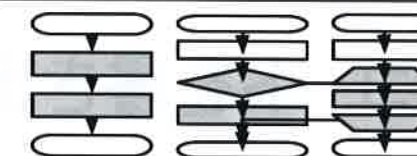
①コードを書き出す

function get_pass_or_failure

```
function get_pass_or_failure(points) {  
  let judge = "合格";  
  for(let i=0; i<number; i++) {  
    if (points[i] < 60) {  
      judge = "不合格";  
      break;  
    }  
  }  
  return judge;  
}  
console.log(get_pass_or_failure(points));
```

ループ処理ステート
が上がる →

②フローチャートをイメージする





DAY02 ワークシート

2月 期氏名: 安井 一

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

https://diver.diveintocode.jp/textbooks/329_newJudgement.js

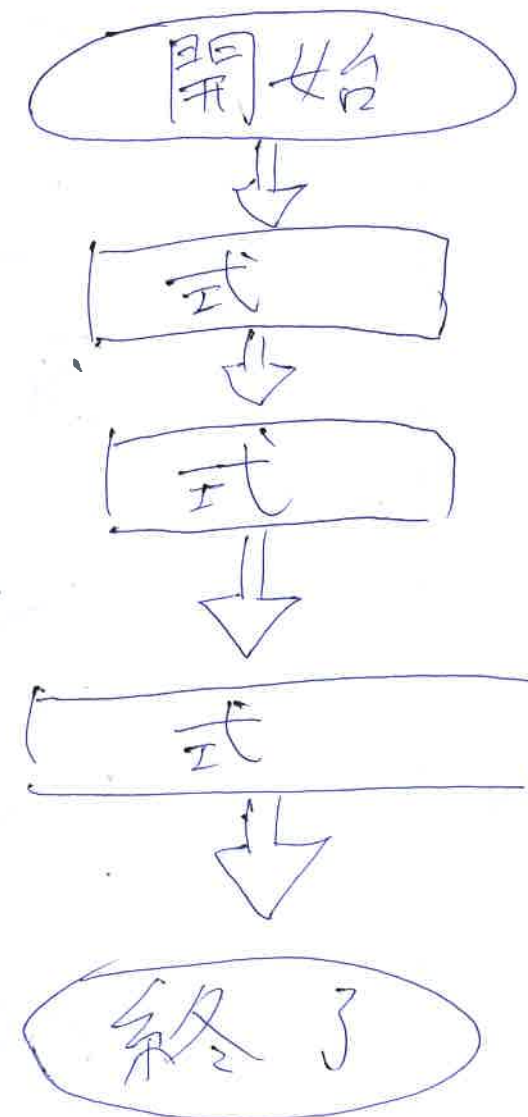
①コードを書き出す

②フローチャートをイメージする



^{judgement}
~~function get_achievement~~

```
function judgement(points){  
  let achievement = get_achievement(points);  
  let pass_or_failure = get_pass_or_failure(points);  
  return `おじいさんの成績は ${achievement} です。 ${pass_or_failure} です。`;  
}  
console.log(judgement(points));
```





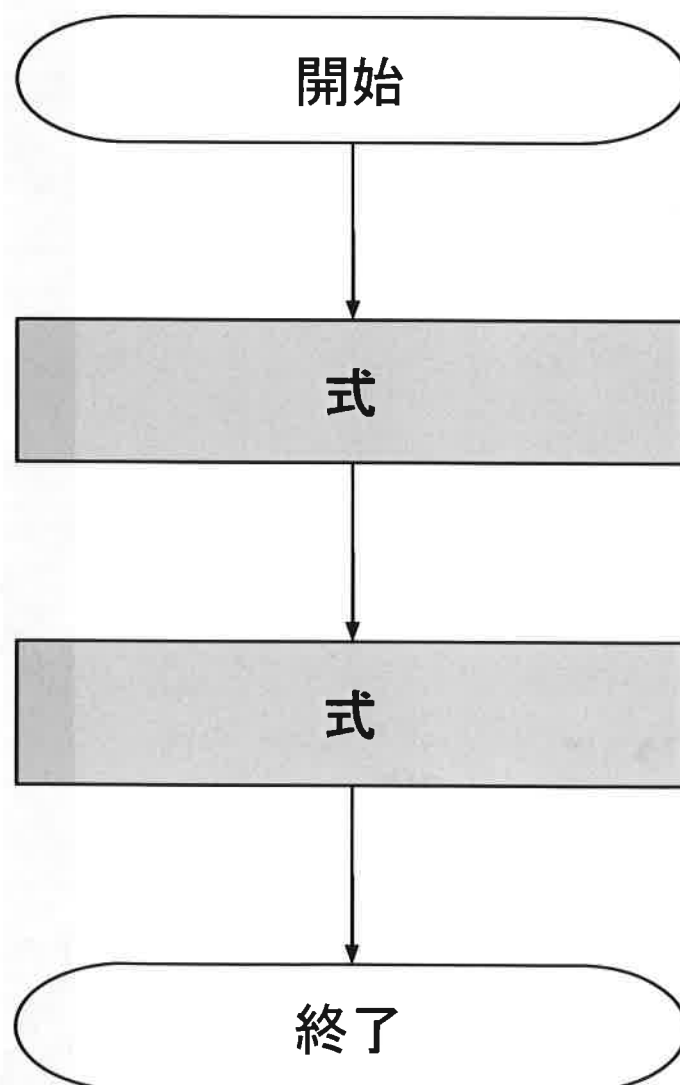
DAY02 ワークシート

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

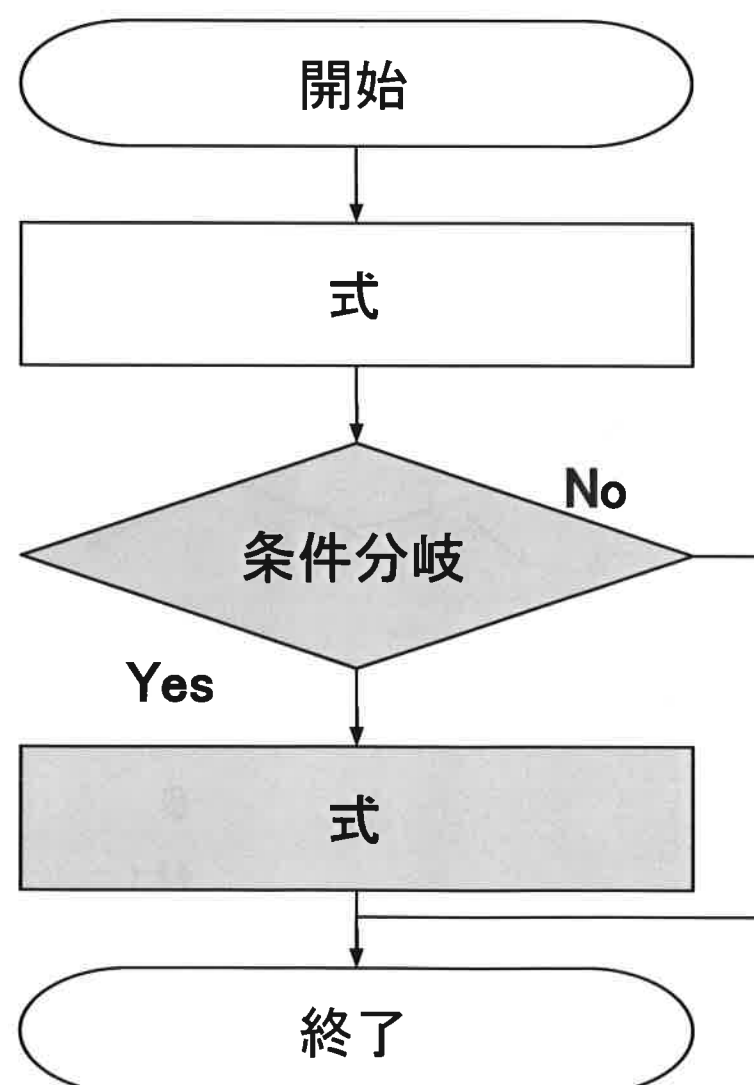
https://diver.diveintocode.jp/textbooks/329_newJudgement.js

フローチャートの書き方は、まず「順序実行、条件分岐、反復(繰り返し)」をおさえておけば大丈夫。

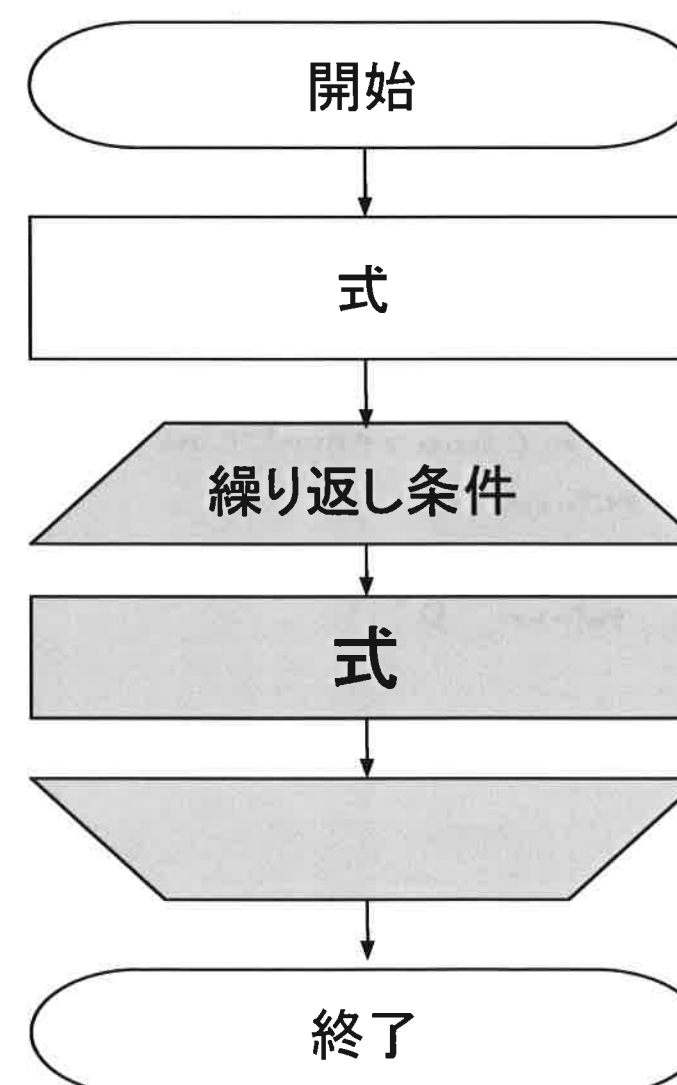
順序実行



条件分岐



反復(繰り返し)





DAY02 ワークシート

18

2月

期氏名:

江口 千太郎

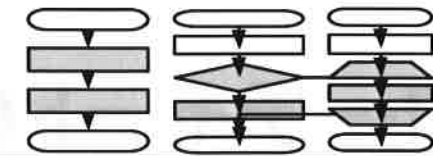
JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

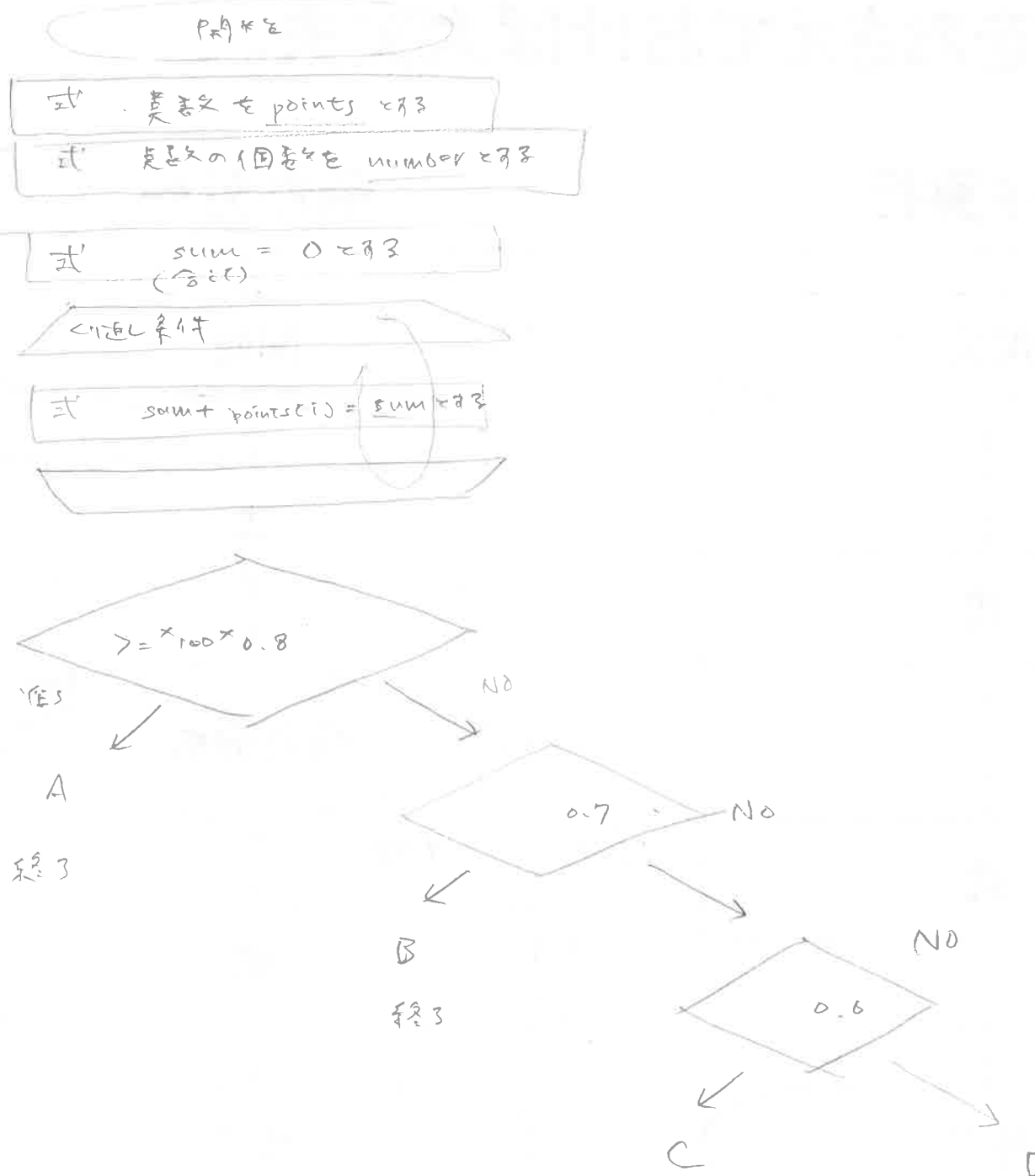
①コードを書き出す

②フローチャートをイメージする



function judgement — get_achievement

```
let points = [60, 80, 100, 80, 90, 80];  
let number = points.length;  
function get_achievement(points) {  
  let sum = 0;  
  for (let i = 0; i < number; i++) {  
    sum = sum + points[i];  
  }  
  if (sum >= number * 100 * 0.8) {  
    return "A"  
  }  
  else if (sum >= number * 100 * 0.7) {  
    return "B"  
  }  
  else if (sum >= number * 100 * 0.6) {  
    return "C"  
  }  
  else {  
    return "D"  
  }  
}
```





DAY02 ワークシート

期氏名:

JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

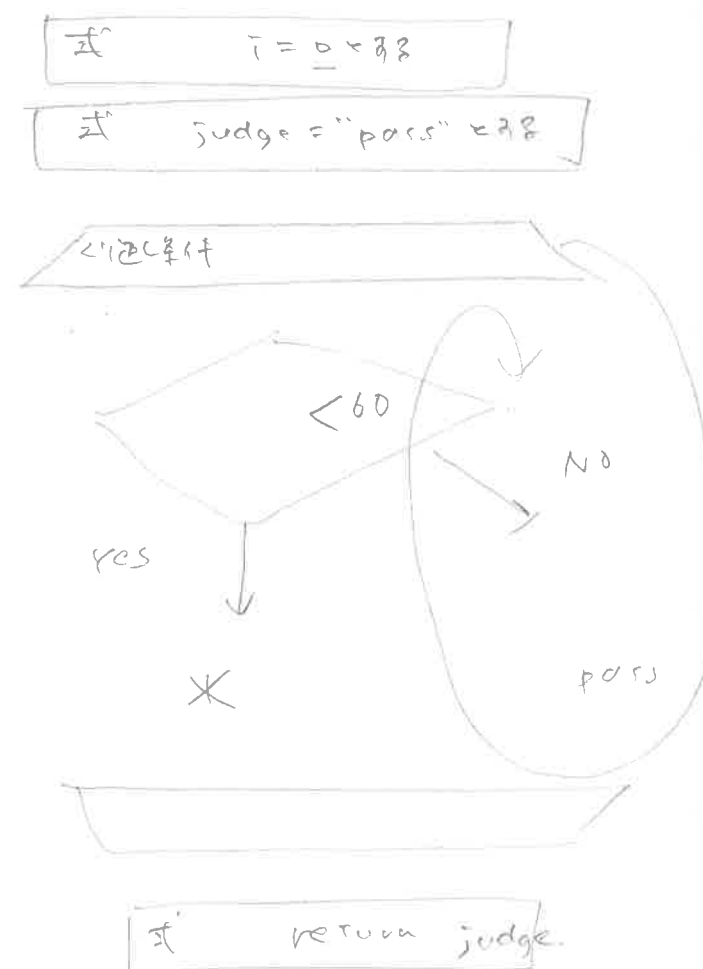
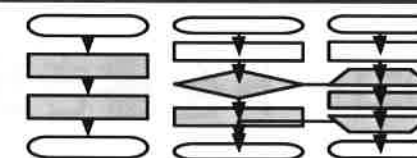
https://diver.diveintocode.jp/textbooks/329_newJudgement.js

①コードを書き出す

function get_pass_or_failure

```
let i = 0;
let judge = "pass";
function get_pass_or_failure(points) {
  for (let i = 0; i < number; i++) {
    if (points[i] < 60) {
      judge = "failure";
    } else (points[i] >= 60)
    { return judge;
  }
}
```

②フローチャートをイメージする





DAY02 ワークシート

18

2 期 氏名: わたなべみなほ

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

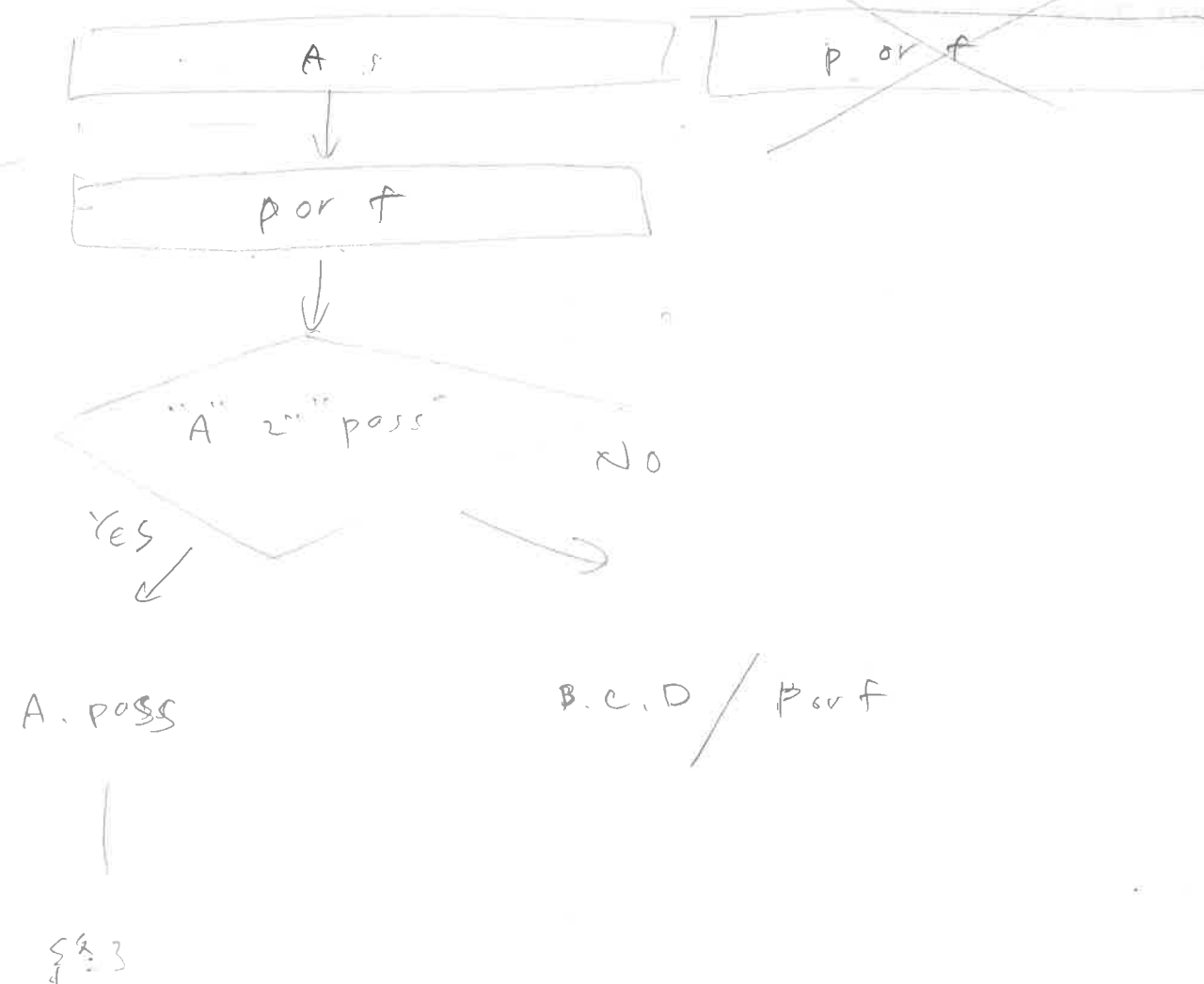
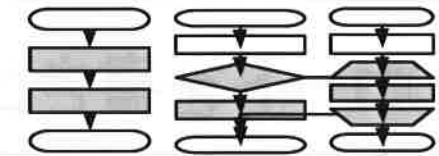
<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

①コードを書き出す

~~function get_achievement~~ judgement

```
function judgement (points) {  
  let achievement = get_achievement (points);  
  let pass_or_failure = get_pass_or_failure (points);  
  if (achievement === A && pass_or_failure === "pass") {  
    return `あなたの成績は ${achievement} です。`  
    // ${pass_or_failure} です。  
  } else {  
    return `あなたの成績は ${achievement} です。`  
    // ${pass_or_failure} です。  
  }  
}
```

②フローチャートをイメージする





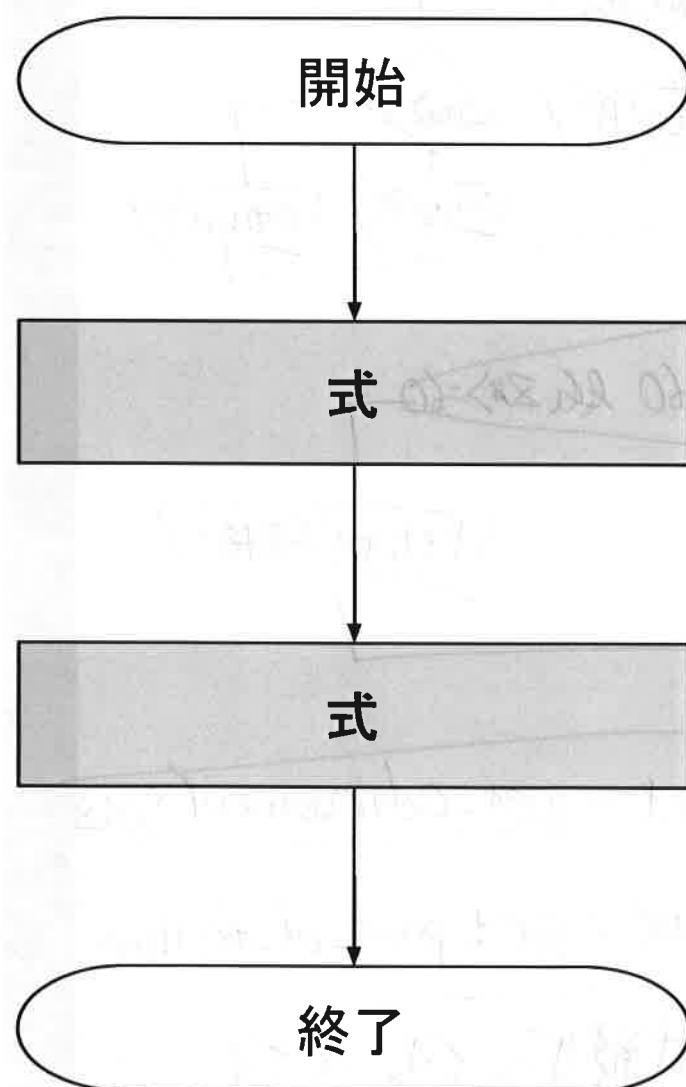
DAY02 ワークシート

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

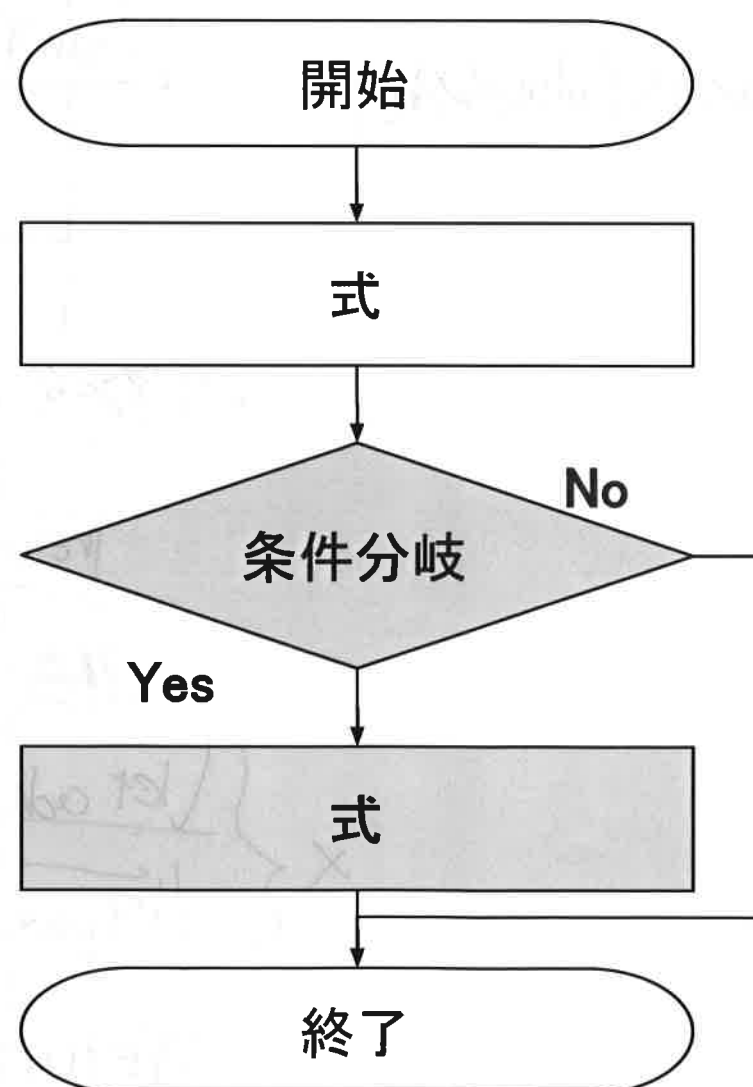
https://diver.diveintocode.jp/textbooks/329_newJudgement.js

フローチャートの書き方は、まず「順序実行、条件分岐、反復(繰り返し)」をおさえておけば大丈夫。

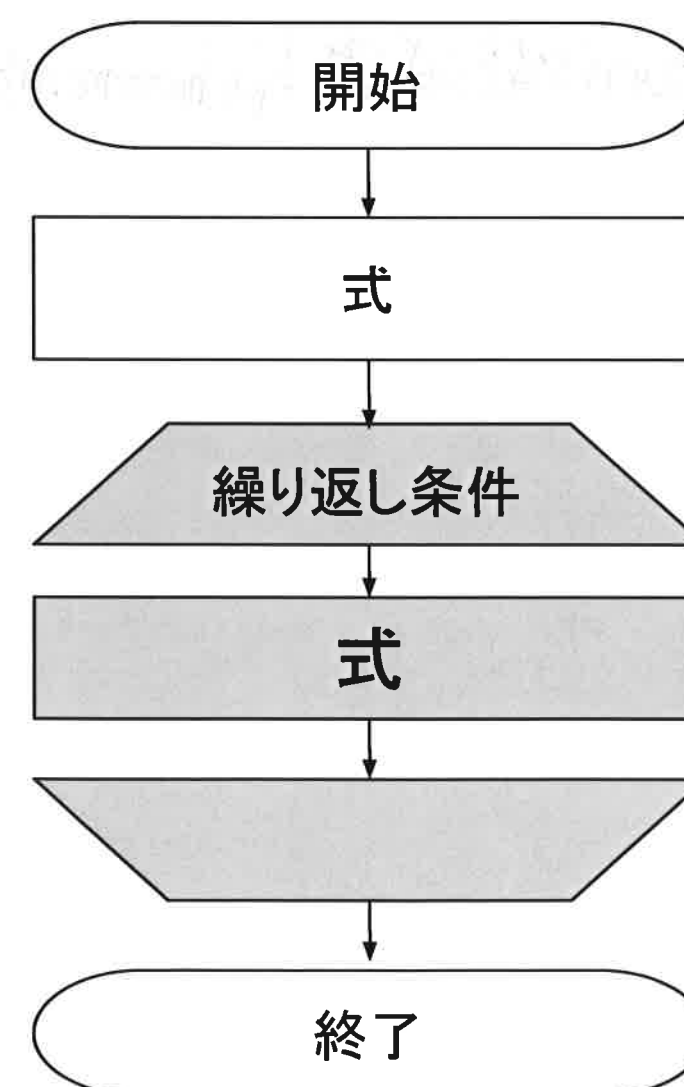
順序実行



条件分岐



反復(繰り返し)





DAY02 ワークシート

2月期氏名: 大門 俊介

JavaScriptシリーズ課題4【合否判定プログラム】

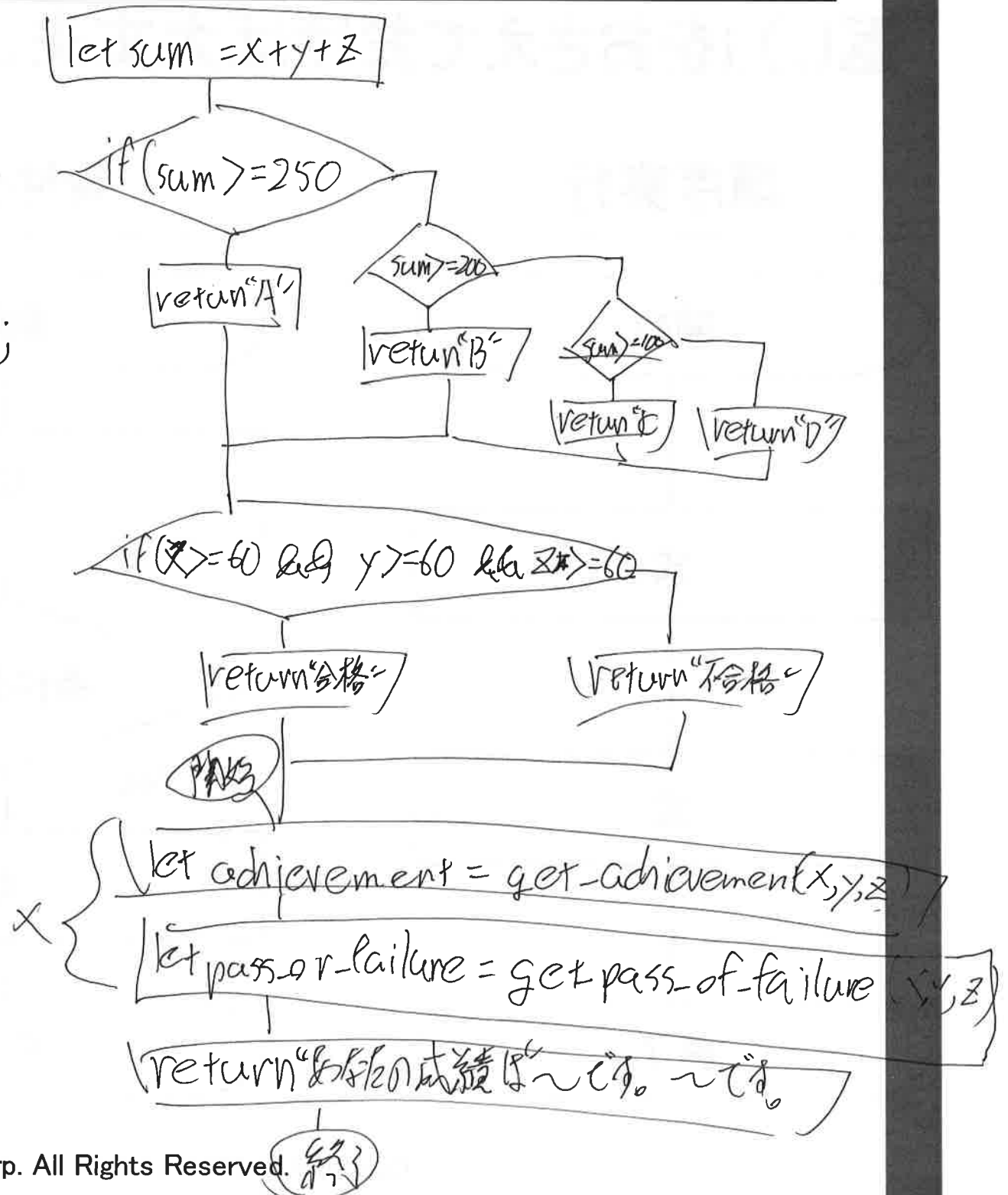
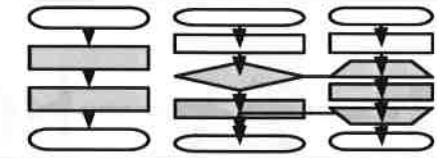
※授業後に回収します

https://diver.diveintocode.jp/textbooks/329_newJudgement.js

①コードを書き出す

```
function judgement (x, y, z) {  
  let achievement = get_achievement(x, y, z);  
  let pass_or_failure = get_pass_or_failure(x, y, z);  
  return `あなたの成績は ${achievement} です。 ${pass_or_failure} です!`;  
}
```

②フローチャートをイメージする





DAY02 ワークシート

2月 期氏名: 大月 俊介

JavaScriptシリーズ課題4【合否判定プログラム】

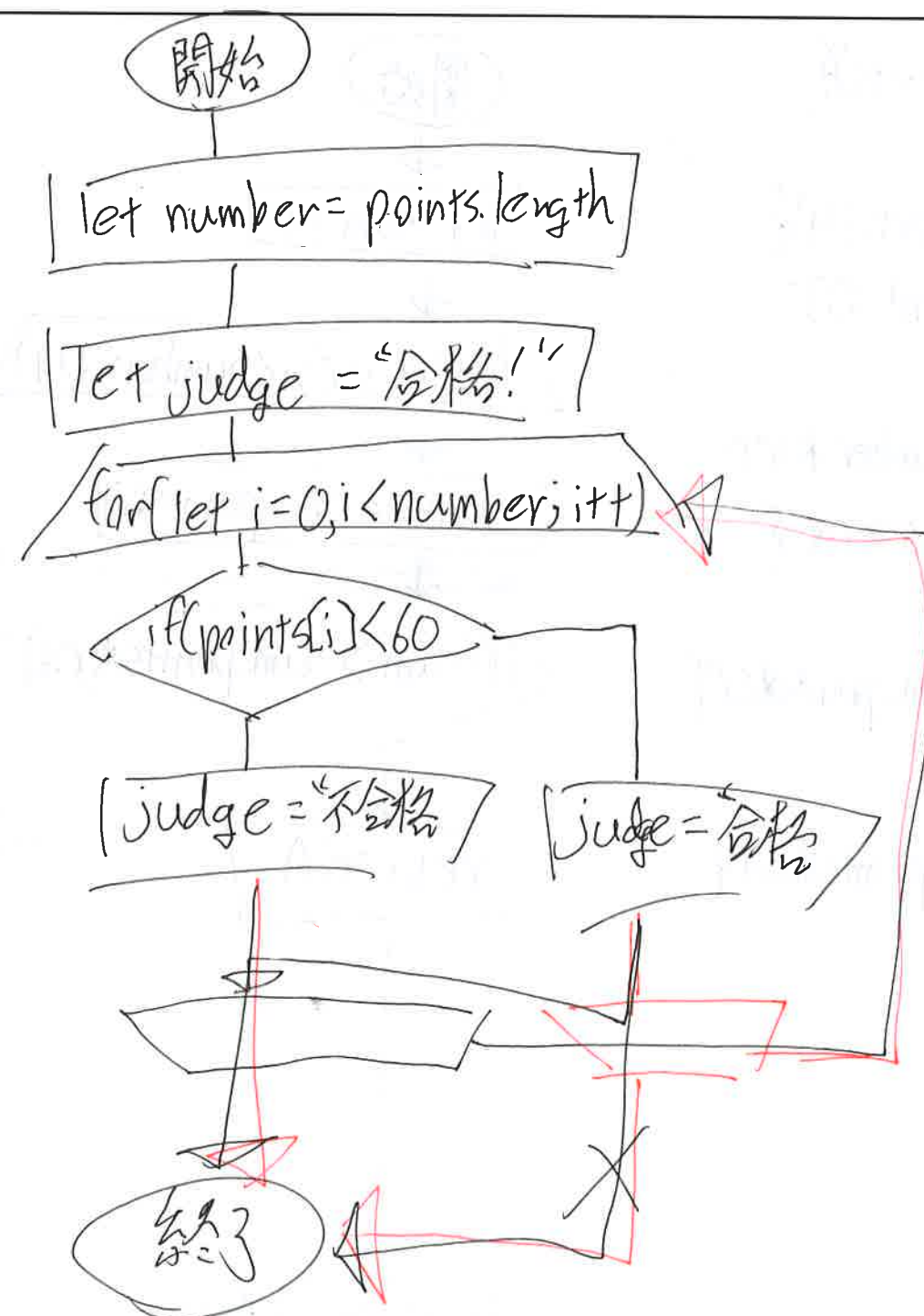
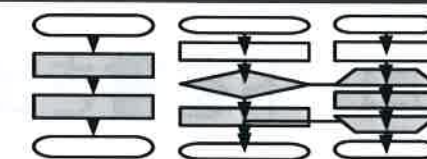
※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329/newJudgement.js>

①コードを書き出す

```
function get_pass_or_failure(points){  
  let number = points.length;  
  let judge = "合格!";  
  for(let i=0; i<number; i++){  
    if(points[i]<60){  
      judge = "不合格";  
      return judge;  
    }  
  }  
  return judge;  
}
```

②フローチャートをイメージする





DAY02 ワークシート

2月 期氏名: 大門 俊介

JavaScriptシリーズ課題4【合否判定プログラム】

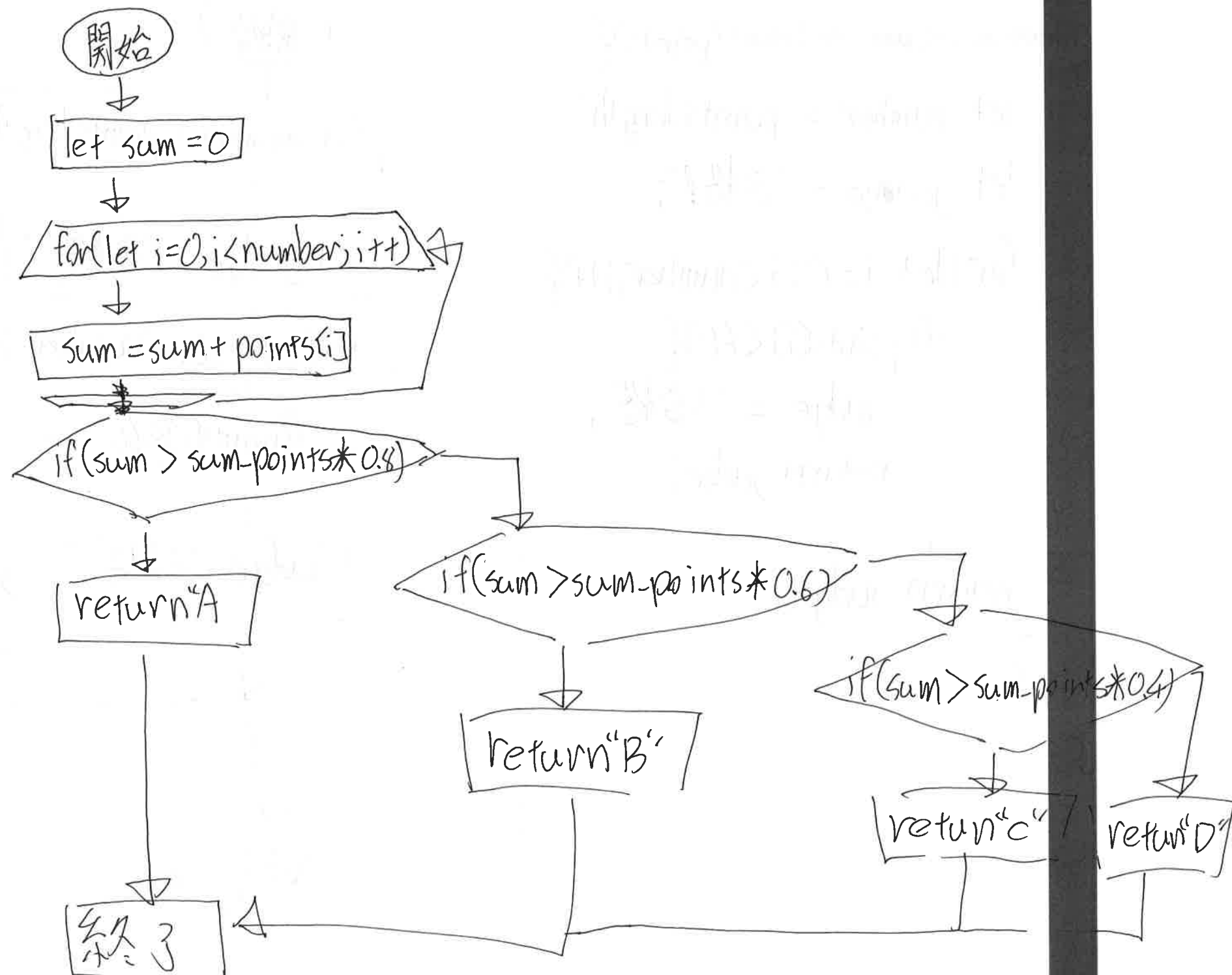
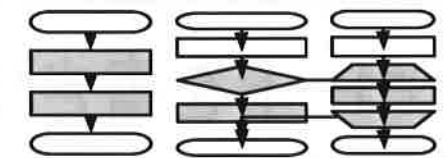
※授業後に回収します

https://diver.diveintocode.jp/textbooks/329_newJudgement.js

①コードを書き出す

```
function get_achievement(points){  
  let sum = 0  
  for (let i=0, i<number; i++){  
    sum = sum + points[i];  
  }  
  let sum_points = number * 100  
  if (sum > sum_points * 0.8){  
    return "A";  
  } else if (sum > sum_points * 0.6){  
    return "B";  
  } else if (sum > sum_points * 0.4){  
    return "C";  
  } else {  
    return "D";  
  }  
};
```

②フローチャートをイメージする





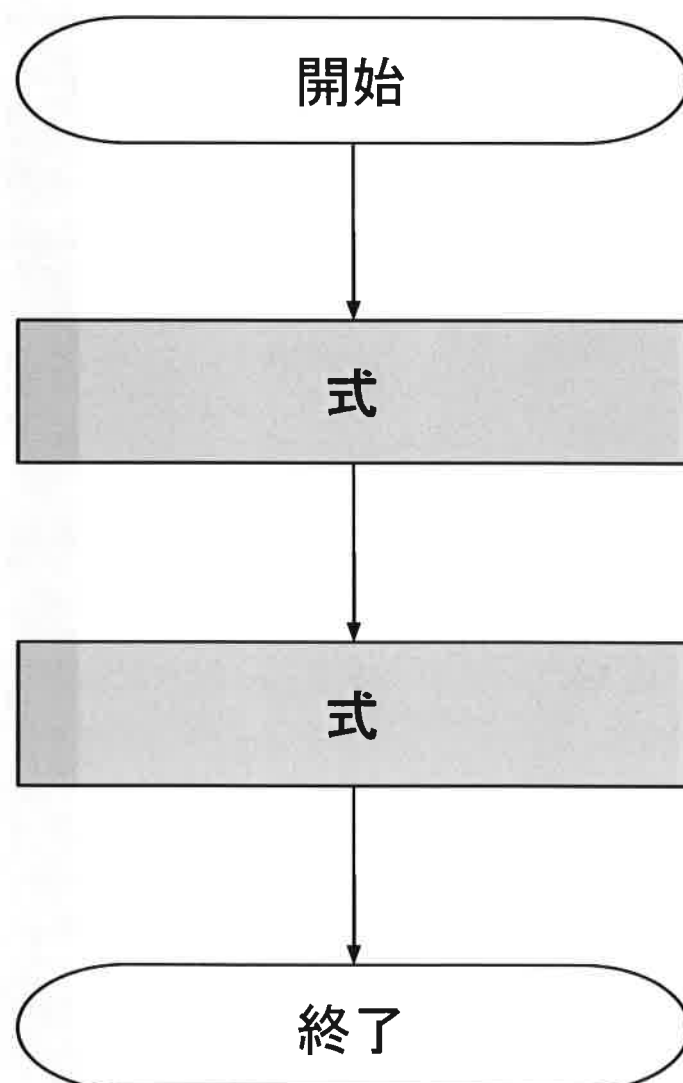
DAY02 ワークシート

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

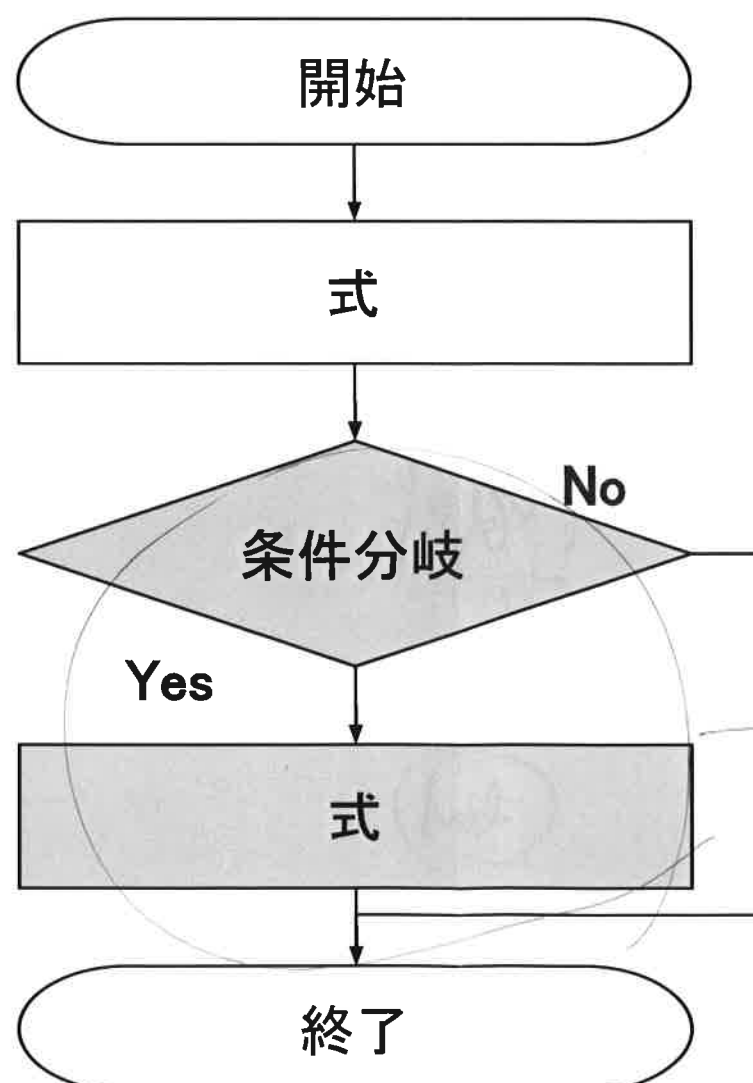
<https://diver.diveintocode.jp/textbooks/329/newJudgement.js>

フローチャートの書き方は、まず「順序実行、条件分岐、反復(繰り返し)」をおさえておけば大丈夫。

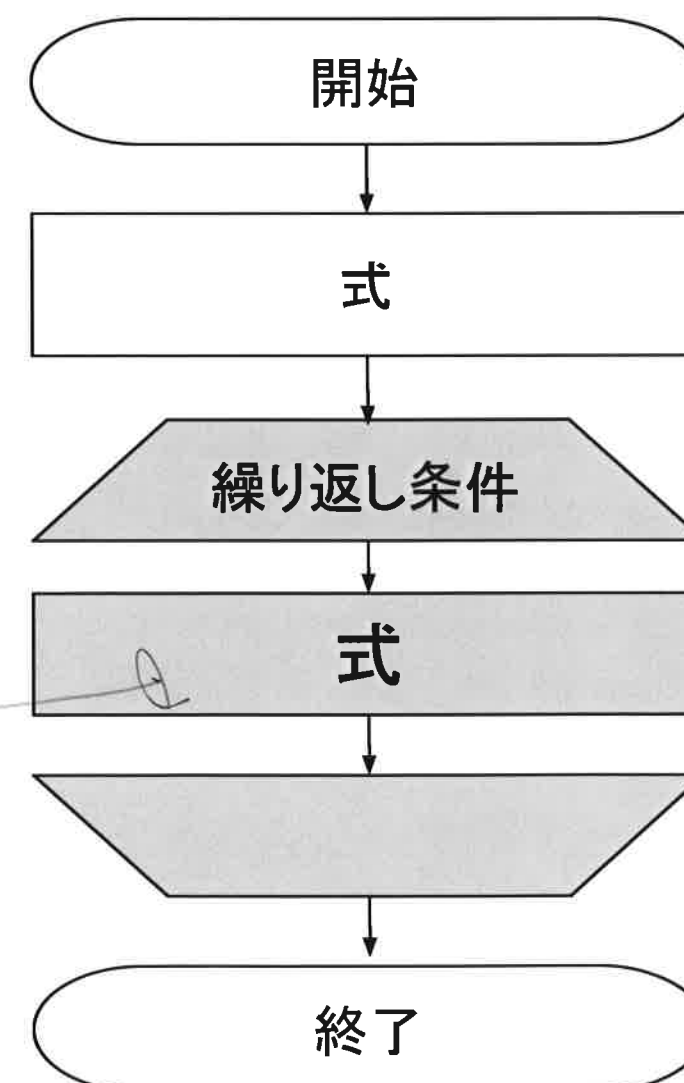
順序実行



条件分岐



反復(繰り返し)





DAY02 ワークシート

期氏名: 中井 示 泰

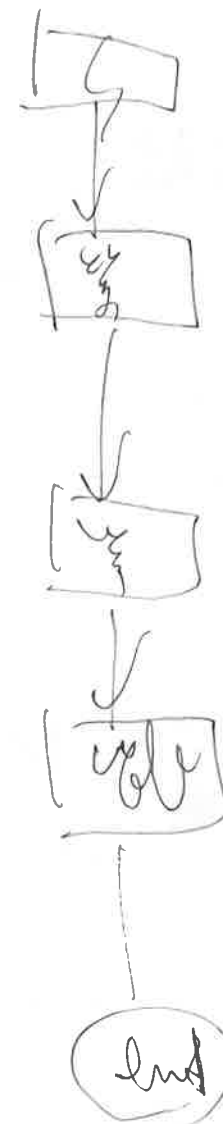
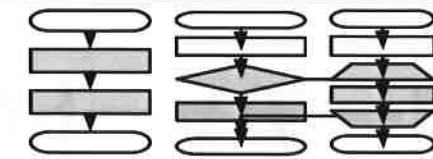
JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

①コードを書き出す

function judgement

②フローチャートをイメージする





DAY02 ワークシート

期氏名:

吉村 和春

JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

①コードを書き出す

function get_pass_or_failure

```
let points = [60, 100, 50, 80, 60, 80]
```

```
let number = points.length
```

```
let sum = 0
```

```
for (let i = 0; i < number; i++)
```

```
  sum += points[i]  
  console.log
```

```
  if (sum >= (number * 100 * 0.4)) {  
    return "C"
```

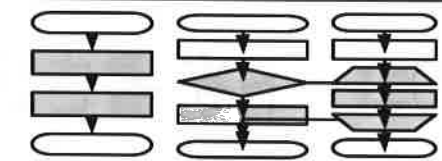
```
  }  
  else {  
    return "D"
```

```
  }  
  console.log(points.length)
```

```
  console.log(get_achievement(points))
```

```
function get_pass_or_failure(points) {
```

②フローチャートをイメージする



```
let judge = "合格"
```

```
for (let i = 0; i < number; i++) {
```

```
  if (points[i] < 60)
```

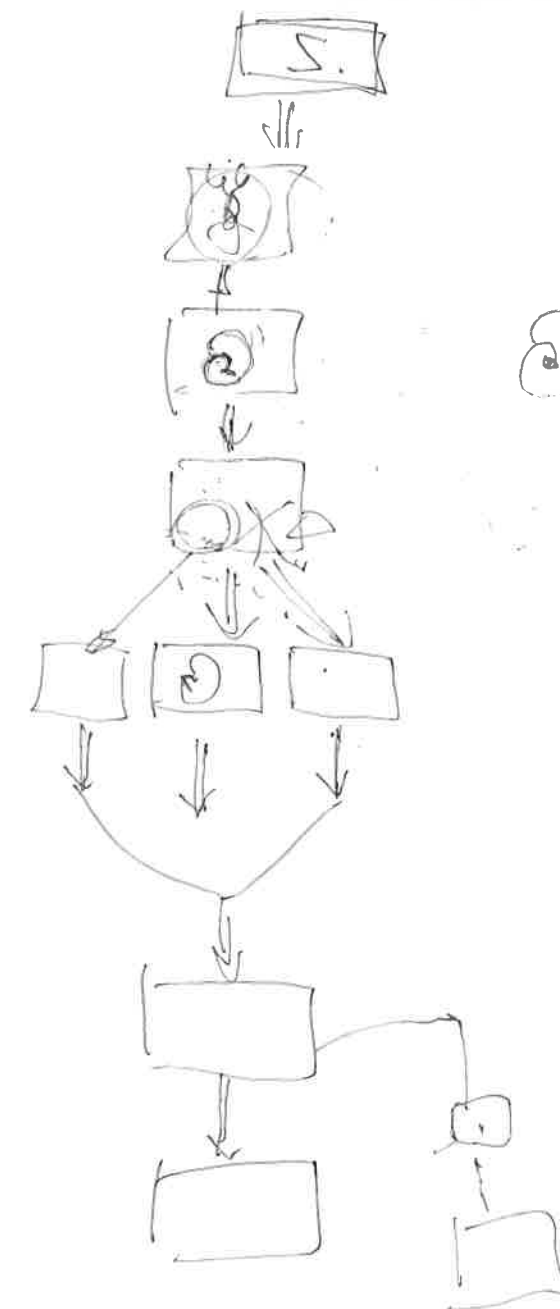
```
    judge = "不合格"
```

```
    break
```

```
    console.log(points[i])
```

```
  }
```

```
  return judge
```





DAY02 ワークシート

期氏名: 志保 和春

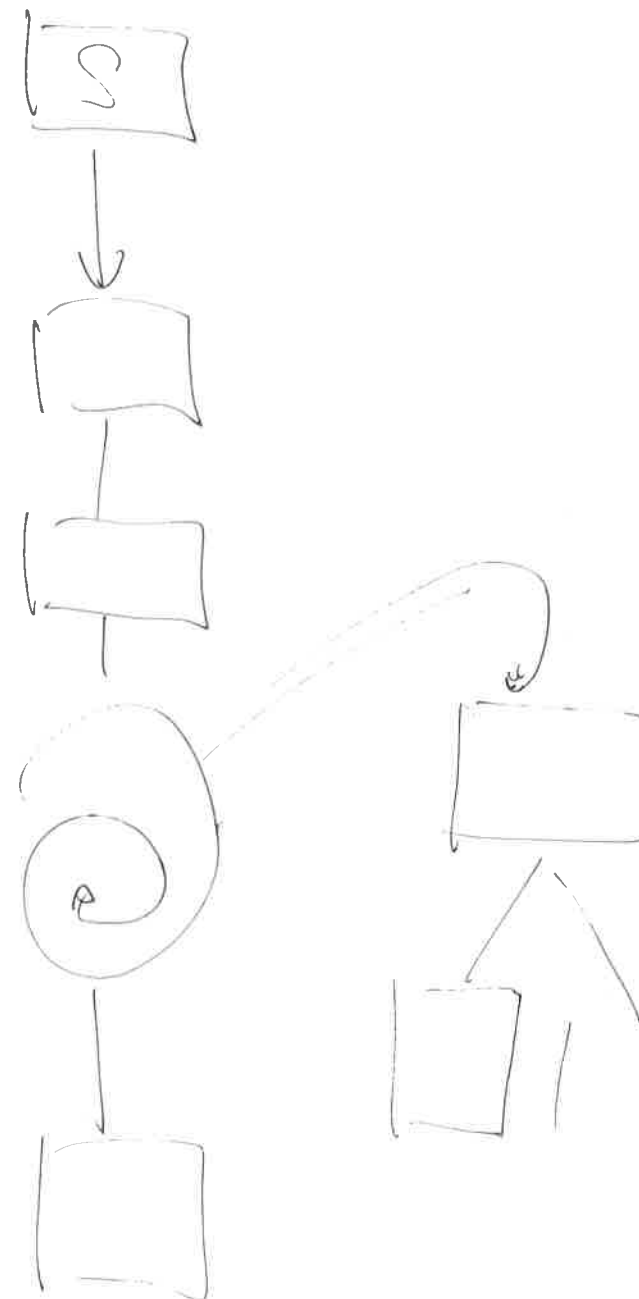
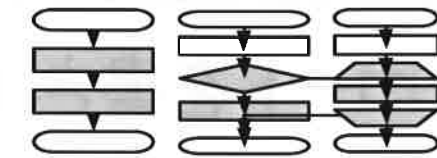
JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

①コードを書き出す

```
function get_achievement
```

②フローチャートをイメージする





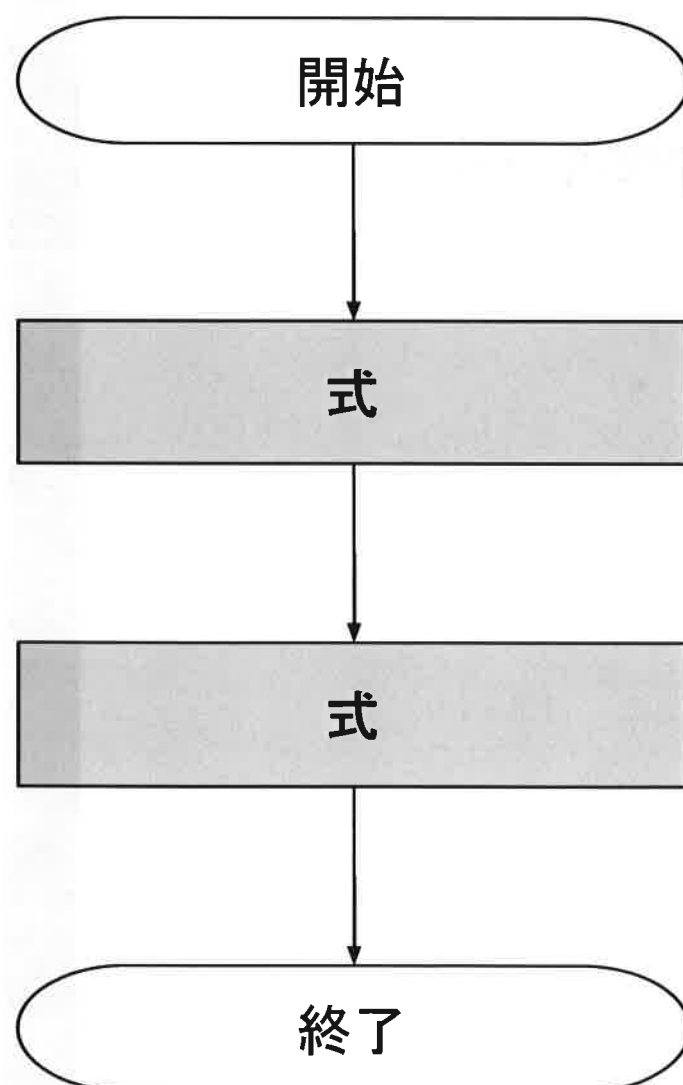
DAY02 ワークシート

JavaScriptシリーズ課題4【合否判定プログラム】 ※授業後に回収します

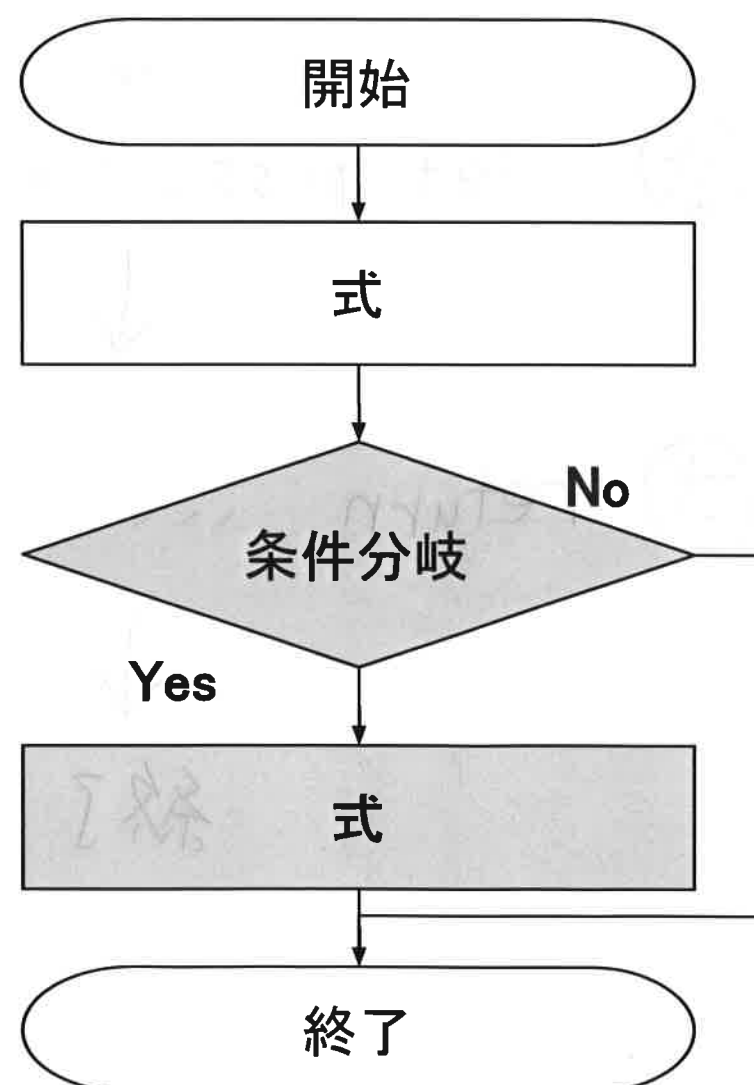
<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

フローチャートの書き方は、まず「順序実行、条件分岐、反復(繰り返し)」をおさえておけば大丈夫。

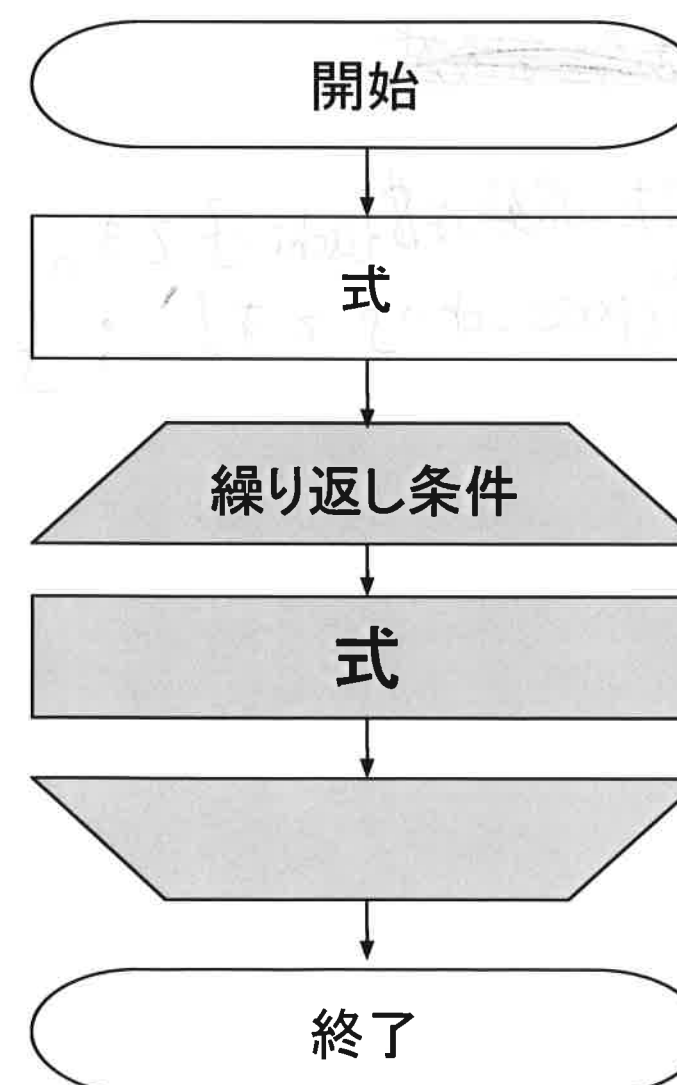
順序実行



条件分岐



反復(繰り返し)





DAY02 ワークシート

2月期 氏名: 金子 健太

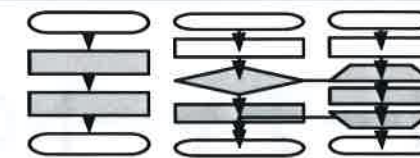
JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

https://diver.diveintocode.jp/textbooks/329_newJudgement.js

①コードを書き出す

②フローチャートをイメージする



```
function judgement (Points)
{
  let achievement = get_achi...(Points)
  let pass_or_... = get_pass...(Points)
  return あなたの成績は
  return あなたの成績は ${achi} です。
    ${pass_or_...} です! ;
}
```

① let ach... = get_achi...



② let pass_... = get_pass...



③ return ...



終了



DAY02 ワークシート

2月期氏名: 金子 健太

JavaScriptシリーズ課題4【合否判定プログラム】

※授業後に回収します

<https://diver.diveintocode.jp/textbooks/329> newJudgement.js

①コードを書き出す

```
function get_pass_or_failure (Points)
{
  let judge = "合格"
  for (let i = 0; i < number; i++)
  {
    if (points[i] < 60)
    {
      judge = "不合格";
      break;
    }
  }
  return judge
}
```

②フローチャートをイメージする

