mission 2-01

達成目標 80 秒

式を見て処理順を示せ①

式に含まれる演算子の処理順を書き込んでください。

1 + 2 * 3 ……▶ 1 ②+ 2 ①* 3

1	1 + 2 * 3	2	(1 + 2) * 3 * 4
3	1 * 2 * 3	4	1 + (2 * 3) * 4
5	1 + 2 - 3	6	1 + (2 + 3) * 4
7	1 / 2 + 3 * 4	8	1 + 2 * 3 * 4 + 5
9	1 / 2 * 3 * 4 + 5 * 6 - 7	10	1 * 2 - 3 * (4 + 5) * 6 - 7

mission 2-02

達成目標 100 秒

式を見て計算結果を示せ

式を見て処理順に計算し、結果を書き込んでください。

1 + (2 - 3 * 4) * 5

12

-10

-50

-49

1 1 + (2 * 3) * 4

2 1 / 2 + 3 * 4

3 1 / 2 * 3 + 4 + 5 * 6 - 7

4 1 * 2 - 3 * (4 + 5) * 2 - 7

5 1 + (2 * 3) * (4 + 5) * 2

6 (1 + 2) - 3 * (4 - 5) * 6

mission 2-03

達成目標 50 秒

プログラムを見て変数に印を付けろ

プログラムのコードを見て、変数の部分の下に印を付けてください。

```
normalPrice = 100
sellingPrice = normalPrice * 1.1
print(sellingPrice)
```

1

```
let text = 'Hello';
console.log(text);
```

2

```
let year = 2019;
let wareki = year - 2018;
console.log(year);
console.log(wareki);
```

3

```
let price = 1000;
let quantity = 10;
let sales = price * quantity;
console.log(sales);
```

4

```
let sales = 9980;
let payment = 10000;
let change = payment - sales;
console.log(payment);
console.log(change);
```

mission 2-04

達成目標 40 秒

適切な変数名を選択せよ

変数名として適切なものを選択してください。

①2021 ②year ③2021year

1 ① 文字列 ② text ③ テキスト

2 ① 1st_check ② 1check ③ check1

3 ① fileName ② file/name ③ file_name ④ FileName

4 ① defaultValue ② default ③ def@ult ④ DEFAULT

mission 2-05

達成目標 40 秒

演算子の合成記法の結果を示せ

プログラムのコードを見て、最後に表示される計算結果を書き込んでください。

```
let year = 2000;
year += 21;
console.log(year);  2021
```

1
```
let i = 0;
i += 1;
console.log(i);
```

2
```
let num = 10;
num -= 5;
console.log(num);
```

3
```
let num = 10;
num /= 5;
console.log(num);
```

4
```
let num = 10;
num++;
console.log(num);
```

5
```
let text = '山';
text += '川';
console.log(text);
```

6
```
let price = 1000;
let discount = 100;
price -= discount;
console.log(price);
```


mission 2-06

達成目標 120 秒

エラーの原因を選べ

プログラムのコードを見て、エラーの原因として適切なものを
選択肢から選択してください。

```
let num1 = 4;
let num2 = 2;
let sum = num1 ++ num2;
console.log(sum);
```

エラーメッセージ

Uncaught SyntaxError: Unexpected identifier

① 1行目で作成している変数名が不正

② 3行目で作成している変数名が不正

③ 3行目で不正な位置に変数名を書いている

1

```
const year = 2021;
year++;
console.log(year);
```

エラーメッセージ

Uncaught TypeError: Assignment to constant variable.

① 1行目で作成している変数名が不正

② 2行目で不正な演算を行っている

③ 2行目でconstで宣言した変数に値を再代入している

```
let price = 1500;
let tax = 150;
pric += tax;
console.log(price + '円');
```

2

エラーメッセージ

```
Uncaught ReferenceError: pric is not defined
```

① 1行目で作成している変数名が不正

② 3行目で定義していない変数を使用している

③ 4行目で型の変換が必要

```
let class = 4;
let studentName = '服部 太朗';
console.log(class + '組 ' + studentName);
```

3

エラーメッセージ

```
Uncaught SyntaxError: Unexpected token 'class'
```

① 1行目で作成している変数名が不正

② 2行目で作成している変数名が不正

③ 3行目で不正な演算を行っている

4

```
let 1price = '100円';
let 2price = '200円';
let price_text = '商品1は' + 1price;
price_text += '商品2は' + 2price;
console.log(price_text);
```

エラーメッセージ

Uncaught SyntaxError: Invalid or unexpected token

① 1行目で作成している変数名が不正

② 3行目で不正な演算を行っている

③ 5行目で不正な演算を行っている

5

```
const discount = 200;
let price = 1100;
price -= discount;
console.log('割引後価格', price, '円');
let price = 1000;
price -= discount;
console.log('割引後価格', price, '円');
```

エラーメッセージ

Uncaught SyntaxError: Identifier 'price' has already been declared

① 3行目でconstで宣言した変数に値を再代入している

② 5行目で宣言済みの変数をもう一度宣言している

③ 6行目で定義していない変数を使用している

mission 3-01

達成目標 40 秒

プログラムを見て関数・メソッドに印を付けろ

プログラムのコードを見て、関数・メソッドの名前の下に印を付けてください。


```
console.log(parseInt(prompt()) + 100);
```

1
```
alert('メッセージに表示されます');
console.log('コンソールに表示されます');
```

2
```
let txt = prompt('Input something: ');
console.log(txt, 'was input');
```

3
```
let price = parseInt(prompt('税抜価格を数字で入力: '));
let tax = price * 0.1;
console.log('消費税額:', tax);
```

4
```
let width = parseInt(prompt('四角形の底辺は？ '));
let height = parseInt(prompt('四角形の高さは？ '));
let square = width * height;
console.log('四角形の面積は', square);
```

mission 3-02

達成目標 50 秒

式を見て処理順を示せ②

式に含まれる下線を引かれた演算子・関数・メソッドの処理順を書き込んでください。

```
console.log(prompt() + 'が入力されました');
        ③     ①      ②
```

1
```
'入力結果: ' + prompt()
```

2
```
let square = width * height
```

3
```
let circle = parseInt(prompt('半径は？ ') ** 2 * 3.14)
```

4
```
console.log(parseInt('4' + '2') + 42)
```

5
```
console.log('2倍にすると' + parseInt(prompt('数を入力: ')) * 2)
```


mission 3-03

達成目標 120 秒

式を見て処理順を示せ③

式に含まれる演算子の処理順を書き込んでください（カッコや.は演算子ではありません）。

```
i + 1 < 10 && i < 100
  ①   ②    ④    ③
```

1
```
! true && true
```

2
```
true || ! false && true
```

3
```
(true || ! false) && true
```

4
```
12 < a + i && a + i < 20
```

5
```
text !== password || text === ''
```

6
```
text !== '山' || text !== '川' || text === ''
```

7
```
! a < 16 || ! 65 < a && a < i + 99
```


mission 3-04

達成目標 150 秒

出力結果はtrueかfalseか

プログラムのコードを見て、最後に表示される真偽値を書き込んでください。


```
let txt = 'abc';
console.log(txt === 'abc');
```
true

1
```
console.log(100 >= 100);
```

2
```
console.log(!(100 < 100));
```

3
```
let text = 'password';
console.log(text !== 'password');
```

4
```
let age = 21;
console.log(age >= 20);
```

5
```
console.log(true && false);
```

6
```
console.log(true || false);
```

7
```
let flg = !true && true;
console.log(flg);
```

8
```
let flg = true || false && true;
console.log(flg);
```

```
9  let text = '山';
   console.log(text === '山' || text === '川');
```

```
10 let flg = (true || false) && true;
   console.log(!flg);
```

```
11 let age = 65;
   console.log(age < 18 || 65 < age);
```

```
12 let text = 'ninja';
   console.log(text === 'NINJA');
```

```
13 let score = 72;
   let bestScore = 70;
   console.log(score >= bestScore);
```

```
14 let bmi = 21.5;
   console.log(18.5 <= bmi && bmi < 25);
```

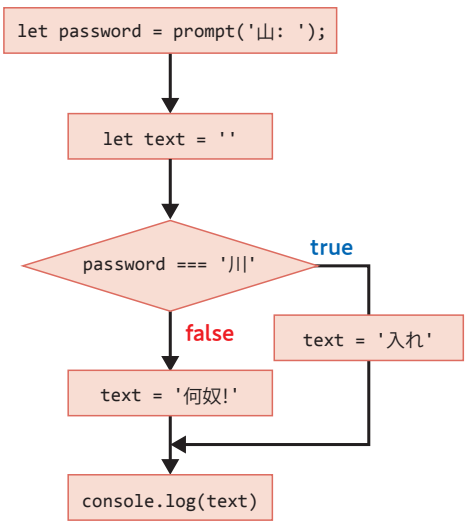
mission 3-05

達成目標 300 秒

コードを見て フローチャートを書け

プログラムのコードを見て、フローチャートを書いてください。


```
let password = prompt('山: ');
let text = '';
if (password == '川') {
    text = '入れ';
}
else {
    text = '何奴！';
}
console.log(text)
```



1

```
let ninja = true;
if (ninja) {
  console.log('にんにん');
}
else {
  console.log('なむなむ');
}
```

2

```
let age = parseInt(prompt());
if (40 <= age) {
  console.log('上忍');
}
else if (30 <= age) {
  console.log('中忍');
}
```

3

```
let mailAddress = true;
let password = false;
if (mailAddress) {
  if (password) {
    console.log('ログイン成功');
  }
  else {
    console.log('ログイン失敗');
  }
}
```

mission 4-01

達成目標 120 秒

出力結果を書け①

プログラムのコードを見て、最後に表示される出力結果を書き込んでください。

```
let irohaArray = [‘い’, ‘ろ’, ‘は’, ‘に’, ‘ほ’, ‘へ’, ‘と’];
console.log(irohaArray[3]);
```

に

1

```
let grade = ['松', '竹', '梅'];
console.log(grade[2]);
```

2

```
let classArray = ['下忍', '中忍'];
classArray.push('上忍');
console.log(classArray);
```

3

```
let planetArray = ['水星', '金星', '地球', '火星', '木星',
                   '土星', '天王星', '海王星', '冥王星'];
let removed = planetArray.pop();
console.log(removed);
```


mission 4-02

達成目標 120 秒

出力結果を書け②

プログラムのコードを見て、最後に表示される出力結果を書き込んでください。

```
let alphabets = ['a', 'b', 'c', 'd', 'e', 'f', 'g'];
console.log(alphabets.length);
```

7

1

```
const week = ['日', '月', '火', '水', '木', '金', '土'];
console.log(week.length);
```

2

```
const band1 = ['John', 'Paul', 'George', 'Ringo'];
const band2 = ['George', 'Andrew'];
console.log(band1.includes('George')
            && band2.includes('George'));
```

3

```
let days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
console.log(days.lastIndexOf(30) - days.indexOf(30));
```

```
4 let tasks = ['掃除', '洗濯', '洗いもの'];
  tasks.fill('済', 0, 2);
  console.log(tasks);
```

```
5 let member = ['久馬', '浅越', '鈴木'];
  member.splice(2, 0, 'ギブソン', 'なだぎ');
  console.log(member);
```

```
6 let member = ['久馬', '浅越', '鈴木'];
  member.splice(2, 0, 'ギブソン', 'なだぎ');
  member.splice(3, 2);
  console.log(member);
```

```
7 let gotairou = ['徳川', '小早川', '前田', '毛利', '宇喜多'];
  let east = gotairou.slice(0, 2);
  console.log(east);
```

```
8 let gotairou = ['徳川', '小早川', '前田', '毛利', '宇喜多'];
  let west = gotairou.slice(2);
  console.log(west);
```


mission 4-03

達成目標 300 秒

出力結果を書け③

プログラムのコードを見て、最後に表示される出力結果を書き込んでください。

```
let animal = '鶴';
let longevity = 1000;
console.log(`${animal}は${longevity}年`);
```

鶴は1000年

1
```
console.log("You say 'why',\nand I say 'I don't know'");
```

2
```
console.log('Jean'.concat('Claude', 'Van', 'Damme'));
```

3
```
let pointArray = [92, 88, 84];
console.log(`${pointArray[1]}点です`);
```


4

```
let point = 90;
console.log(`結果は
${point >= 80 ? '合格' : '不合格'}です`);
```

5

```
let folderPath = String.raw`Users\ninja\Documents`;
let fileTitle = 'memo.txt';
console.log(`${folderPath}\\${fileTitle}`);
```

6

```
let htmlText = '古池や<br>蛙飛びこむ<br>水の音';
console.log(htmlText.replaceAll('<br>', '\n'));
```

7

```
let mailText = '\n\nお世話になっております。\n忍者です\n\n';
console.log(mailText.trim());
```

```
let tel = '03-1234-5678';
if (tel.split('-')[0] === '03') {
  console.log('東京都');
}
else {
  console.log('それ以外');
}
```

8

```
let url = 'https://pub.mynavi.jp';
console.log(`${url.endsWith('.jp') ?
              '日本のドメイン' : 'それ以外'}`);
```

9

```
let flightNumber = 'JL-101';
switch (flightNumber.split('-')[0]) {
  case 'NH':
    console.log('全日空');
    break;
  case 'JL':
    console.log('日本航空');
    break;
  case 'AA':
    console.log('アメリカン航空');
    break;
}
```

10

mission 4-04

達成目標 120 秒

出力結果を書け④

プログラムのコードを見て、最後に表示される出力結果を書き込んでください。

```
let hands = {
  rock : 'beats scissors',
  paper : 'beats rock',
  scissors : 'beats paper',
};
console.log(hands.rock);
```

beats scissors

```
let reviews = {
  '2009': '50年に一度の出来',
  '2010': '新酒らしいフレッシュな味',
  '2011': '21世紀最高の出来',
};
console.log(reviews['2009']);
```


2

```
let scores = {
  math: 98,
  japanese: 70,
  science: 89,
};
console.log(`数学:${scores['math']}点`);
```

3

```
let jackson5 = {
  Jackie: 'tenor', Tito: 'baritone',
  Jermaine: 'bass', Marlon: 'percussion',
  Michael: 'lead',
};
console.log('Randy' in jackson5);
```

3

```
let scores = {
  math: 98,
  japanese: 70,
  science: 89,
};
if (scores.math) {
  console.log('数学受験済');
}
else {
  console.log('数学未受験または0点');
}
```

mission 5-01

達成目標 300 秒

出力結果を書け⑤

プログラムのコードを見て、最後に表示される出力結果を書き込んでください。

```
targets = ['マトA', 'マトB', 'マトC']
for (let target of targets) {
  console.log(`${target}に手裏剣を投げた`)
};
```

マトAに手裏剣を投げた
マトBに手裏剣を投げた
マトCに手裏剣を投げた

1

```
for (let i = 0; i < 3; i++) {
  console.log('bicycle');
}
```

2

```
for (let i = 0; i < 4; i++) {
  console.log(i ** 2);
}
```

3

```
let pointList = [92, 88, 84];
for (let point of pointList) {
  console.log(`${point}点です`);
}
```

4

```
for (let i = 0; i < 3; i++) {
  if (i === 1) {
    console.log('ああ松島や');
  }
  else {
    console.log('松島や');
  }
}
```

5

```
let givenNames = ['Alan', 'Ryan'];
let familyNames = ['Moore', 'Scott'];
for (let givenName of givenNames) {
  for (let familyName of familyNames) {
    console.log(givenName, familyName);
  }
}
```


mission 5-02

達成目標 60 秒

どの構文を使うのが最適かを選べ

問題文に書かれたような処理を行いたい場合、
どの構文を使うのが最適か選択してください。

処理：手裏剣を3回投げる

① **for文による繰り返し**
② **for-of文による繰り返し**
③ **while文による繰り返し**

1

処理：孔明を3回訪ねる

① **for文による繰り返し**
② **for-of文による繰り返し**
③ **while文による繰り返し**

2

処理：孔明が「はい」と言うまで訪ねる

① **for文による繰り返し**
② **for-of文による繰り返し**
③ **while文による繰り返し**

3

処理：孔明がいる可能性のある場所を1つずつ順番に訪ねる

① **for文による繰り返し**
② **for-of文による繰り返し**
③ **while文による繰り返し**

mission 5-03

達成目標 60 秒

出力結果を書け⑥

プログラムを見て、表示される出力結果を書き込んでください。

```
let inPocket = 3500;
while (300 <= inPocket) {
  console.log(inPocket);
  inPocket -= 1000;
}
```

```
3500
2500
1500
500
```

1

```
let total = 13;
while (total <= 21) {
  console.log(total);
  total += 3;
}
```

2

```
let number = 2;
do {
  console.log(number);
  number *= 2;
}
while (number <= 50)
```


mission 6-01

達成目標 150 秒

行の処理順を書け

プログラムを見て、行が処理される順番を書き込んでください
（厳密にはアロー関数式またはfunction文による関数名の登録が先に行われますが、
この問題では①のところから開始とみなしてください）。

```
        let shiftOrder = (name) => {
④②       console.log(`${name}様`);
        }


①       shiftOrder('服部');
③       shiftOrder('河村');
```

1

```
    let exclaim = (word) => {
      console.log(`${word}!`);
    }

  ①exclaim('にんにん');
```

2

```
    let addTax = (amount, taxRate) => {
      return amount * (1.0 + (taxRate / 100));
    }

  ①let price = 1100;
    console.log(`税込価格${addTax(price, 10)}円`);
```

3

```
  let introduceSelf = (born, grown) => {
    console.log(`${born}生まれ ${grown}育ち`);
  }

①introduceSelf('東京', 'HIP HOP');
  introduceSelf('大阪', 'J-POP');
```

4

```
  let callName = (patient) => {
    return `${patient}様　診察室へお入りください。`;
  }

①let patients = ['磯野', '波野'];
  for (patient of patients) {
    console.log(callName(patient))
  }
```

5

```
  let calculateTriangle = (base, height) => {
    return base * height /2;
  }

  let outputTriangle = (base, height) => {
    let area = calculateTriangle(base, height);
    console.log(`底辺${base}cm、高さ${height}cmの三角形は
${area}cm²`);
  }

①outputTriangle(5, 10);
```


mission 6-02

達成目標 100 秒

出力結果を書け⑦

プログラムを見て、表示される出力結果を書き込んでください。

```
let wokashi = (spring, summer, fall, winter) => {
  console.log(`春は${spring} 夏は${summer}
秋は${fall} 冬は${winter}`);
}

wokashi('あけぼの', '夜', '夕暮れ', 'つとめて')
```

春はあけぼの 夏は夜
秋は夕暮れ 冬はつとめて

1

```
let wagahai = (species = '猫') => {
  console.log(`吾輩は${species}である`);
}

wagahai('犬');
```


2

```
let getAverage = (...numbers) => {
  let sum = 0;
  for (n of numbers) {
    sum += n;
  }
  return sum / numbers.length;
}

console.log(getAverage(10, 15, 20));
```

3

```
let filterNames = (...names) => {
  let result = names.filter(name => name.includes('家'));
  console.log(`家が付くのは${result.length}人`);
}

filterNames('家康', '秀忠', '家光', '家綱', '綱吉');
```

mission 7-01

達成目標 60 秒

オブジェクトについて正しい説明を3択から選べ

オブジェクトの説明として正しいものに丸を付けてください。

1

① オブジェクトはクラスで設計しなければならない。
② オブジェクトはプロパティの集まりである。
③ オブジェクトリテラルは正しいオブジェクトの定義方法ではない。

2

① DateオブジェクトはDOM APIの一種である。
② DateオブジェクトはNode.jsでも利用できる。
③ 標準組み込みオブジェクトは追加インストールが必要である。

3

① new演算子を省略しても新しいオブジェクトを作成できる。
② Dateオブジェクトは1970年1月1日以前の日付を記録できない。
③ 日本標準時はUTCより9時間進んでいる。

mission 7-02

達成目標 120 秒

出力結果を書け⑧

プログラムの結果を予想して書いてください。

```
let obj = {
  lastname: 'yamada',
  getSanDuke() {
    return `${this.lastname}さん`;
  }
}

console.log(obj.getSanDuke());
```

yamadaさん

1

```
let obj = {
  firstname: 'taro',
  lastname: 'yamada',
  getFullName() {
    return `${this.firstname} ${this.lastname}`;
  }
}

obj.firstname = 'jiro';
console.log(obj.getFullName());
```

2

```
let obj = {
  taxrate: 0.1,
  getPrice(value) {
    return value * (1 + this.taxrate);
  }
}

console.log(obj.getPrice(1000));
```

3

```
let obj = {
  taxrate: 0.1,
  getPrice(value) {
    taxrate = 0.2;
    return value * (1 + this.taxrate);
  }
}

console.log(obj.getPrice(1000));
```

4

```
let obj = {
  getSanDuke(name) {
    this.lastname = name;
    return `${this.lastname}さん`;
  }
}

console.log(obj.getSanDuke());
console.log(obj.getSanDuke('sato'));
```


mission 8-01

達成目標 60 秒

指定した要素を取得する CSSセレクターを書け

querySelectorメソッドでHTMLファイルの下線の箇所を取得するための
CSSセレクターを書き込んでください。

```
<body>
  <h1 class="chapter_name">HTMLを操作する</h1>      例
  <div id="text">この文字列を変更せよ</div>      ①
  <p>忍者です。</p>      ②
  <p>HTMLを勉強中です。</p>
  <p class="tel">電話番号は〇〇です。</p>      ③
  <script src="JavaScript.js"></script>
</body>
```

例
```
querySelector('.chapter_name')
```

1
```
querySelector('            ')
```

2
```
querySelector('            ')
```

3
```
querySelector('             ')
```

mission 8-02

達成目標 40 秒

メソッドで取得できる要素に下線を引け

querySelectorメソッド、querySelectorAllメソッドで取得される要素に
下線を引いてください。

```
querySelector('.chapter_name')
```

```
<body>
  <h1 class="chapter_name">HTMLを操作する</h1>
  <div id="text">この文字列を変更せよ</div>
  <p>忍者です。</p>
  <script src="JavaScript.js"></script>
</body>
```

1

```
querySelector('.chapter_name')
```

```
<div class="chapter_name">HTMLを操作する</div>
<div class="chapter_name">JavaScriptの新しい構文</div>
<p>忍者です。</p>
```

2

```
querySelectorAll('p')
```

```
<p class="chapter_name">HTMLを操作する</p>
<p id="text">この文字列を変更せよ</p>
<p>忍者です。</p>
```

mission 8-03

達成目標 70 秒

新しい要素が追加される場所に線を引け

JavaScriptが実行された結果、新しい要素newElementが追加される場所に線を引いてください。

JavaScript.js

```
let parent = document.body;

let newElement = document.createElement('p');
parent.appendChild(newElement);
```

HTML

```
 <body>
   <h1 class="chapter_name">HTMLを操作する</h1>
   <div id="text_div">
     <p>長男</p>
     <p>次男</p>
   </div>
   <script src="JavaScript.js"></script>

  </body>
```

1

JavaScript.js

```
let parent = document.querySelector('#text_div');
let newElement = document.createElement('p');
parent.appendChild(newElement);
```

HTML

```
<body>
  <div id="text_div">
    <p>長男</p>
  </div>
  <script src="JavaScript.js"></script>
</body>
```

2

JavaScript.js

```
let reference = document.querySelector('p');
let parent = reference.parentElement;
let newElement = document.createElement('p');
parent.insertBefore(newElement, reference);
```

HTML

```
<body>
  <div id="text_div">
    <p>次男</p>
    <p>三男</p>
  </div>
  <script src="JavaScript.js"></script>
</body>
```

3

JavaScript.js

```
let parent = document.querySelecor('.text_div');
let newElement = document.createElement('p');
parent.appendChild(newElement);
```

HTML

```
<body>
  <div class="text_div">
    <p>長男</p>
  </div>
  <div class="text_div">
    <p>長女</p>
  </div>
  <script src="JavaScript.js"></script>
</body>
```


mission 8-04

達成目標 60 秒

正しいイベントを選択せよ

次のようなプログラムを書くときに、処理を登録するべきイベントを選択してください。

ボタンがクリックされたときに処理を行う

① clickイベント
② dblclickイベント
③ mouseoutイベント

1

画像がダブルクリックされたら、拡大表示する

① keydownイベント
② mousedownイベント
③ dblclickイベント

2

マウスポインタが動いている間、マウスポインタの軌跡を表示する

① mouseoverイベント
② mouseupイベント
③ mousemoveイベント

3

入力フォームが正常に送信されたときにメッセージを表示したい

① **submitイベント**
② **clickイベント**
③ **loadイベント**

4

パスワード入力欄にキーが入力されるたびに入力されたパスワードの強度を表示する

① **clickイベント**
② **keydownイベント**
③ **mouseoverイベント**

5

要素にフォーカスが当たったときとフォーカスが外れたときにCSSクラスを変更する

① **loadイベントとunloadイベント**
② **focusイベントとblurイベント**
③ **mouseoverイベントとmouseoutイベント**

mission 9-01

達成目標 120 秒

出力結果を書け⑨

プログラムの結果を予想して書いてください。

```
let result = document.querySelector('#result');
let url = '/chap9/test.json';
fetch(url)
  .then((response) => {
    return 'ホップ';
  })
  .then((data) => {
    console.log(data);
  });
```

ホップ

```
let result = document.querySelector('#result');
let url = '/chap9/test.json';
fetch(url)
  .then((response) => {
    return 'ホップ';
  })
  .then((data) => {
    console.log(data);
    return data + 'ステップ';
  })
  .then((data) => {
    console.log(data);
    return data + 'ジャンプ';
  })
  .then((data) => {
    console.log(data);
  });
```

1

2

```
async function getTestJSON(url) {
  let response = await fetch(url);
  let data = await response.json();
  text = '取得データ:' + data['text'];
  console.log(text);
}

getTestJSON('/chap9/test.json');
```

3

```
async function getTestJSON(url) {
  let response = fetch(url);
  console.log(response.json);
}

getTestJSON('/chap9/test.json');
```

※これらのサンプルはc9_2_1.jsやc9_2_2.jsと同様に、Webサーバーや HTML、JSON ファイルが用意された状態で実行されるものとします。test.jsonの内容も同一とします（P.189参照）。

mission 9-02

達成目標 60 秒

正しい説明を3択から選べ

正しいものに丸を付けてください。

1

① asキーワードを使うとエクスポートされていない関数をインポートできる。
② プログラム全体を関数定義で囲むとモジュールになる。
③ export/importはWebサーバーを介して利用する。

2

① 残余引数とスプレッド構文の働きは同じである。
② 残余引数で配列を連結できる。
③ スプレッド構文で配列を連結できる。

3

① オプショナルチェーン演算子はプロパティには使えるが、メソッドには使えない。
② Null合体演算子は、オブジェクトリテラルのプロパティの短縮表記である。
③ Null合体演算子は文字列以外を返すことができる。

mission 10-01

達成目標 200 秒

ドキュメントの説明文の意味を選べ

MDN Web Docsからの引用文を見て、
その意味を正しく表している文に○を付けてください。

1

「for...of」より引用
for...of 文は、反復可能オブジェクト、たとえば組込みの String, Array, 配列状オブジェクト（例えば arguments や NodeList), TypedArray, Map, Set, およびユーザー定義の反復可能オブジェクトなどに対して、反復的な処理をするループを作成します。これはオブジェクトのそれぞれの識別可能なプロパティの値に対して、実行される文を表す独自の反復フックを呼び出します。
https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Statements/for...of

① すべてのオブジェクトは反復可能オブジェクトである

② NodeListは反復可能オブジェクトである

③ argumentsは配列である

2

「String」より引用
JavaScript では、String オブジェクトとプリミティブ文字列は区別されることに注意してください。(Boolean や Number にも同じことが言えます。)
文字列リテラル（二重引用符または単一引用符で示されます)、および String 関数をコンストラクター以外の場面で（すなわち new キーワードを使わずに）呼び出した場合はプリミティブの文字列になります。JavaScript では、必要に応じてプリミティブの文字列が自動的に String オブジェクトに変換されるので、プリミティブの文字列に対して String オブジェクトのメソッドを使用することができます。プリミティブの文字列に対して、メソッドの呼び出しやプロパティの参照が行われようとした場合、JavaScriptは自動的にプリミティブの文字列をオブジェクトでラップし、メソッドを呼び出したりプロパティの参照を行ったりします。
https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Global_Objects/String

① 文字列リテラルとプリミティブ文字列は同じものである

② プリミティブ文字列でもStringオブジェクトのメソッドを呼び出せる

③ String関数はnewキーワードを省略できない

3

「EventTarget.addEventListener()」より引用
options 省略可
対象のイベントリスナーの特性を指定する、オプションのオブジェクトです。次のオプションが使用できます。
capture
Boolean 値で、この型のイベントが DOM ツリーで下に位置する EventTarget に配信される前に、登録された listener に配信されることを示します。
https://developer.mozilla.org/ja/docs/Web/API/EventTarget/addEventListener

① このメソッドはoptions型のオブジェクトを使用できる

② optionsにはオブジェクトを指定できる

③ captureという名前の引数を指定できる

4

「Response.json()」より引用
返値
JavaScript オブジェクトに解決される Promise。このオブジェクトは、オブジェクト、配列、文字列、数値など、JSON で表現できるものであれば何でもなります。
https://developer.mozilla.org/ja/docs/Web/API/Response/json

① メソッドの戻り値はJavaScript型のオブジェクトである

② JSON形式の文字列を取得できる

③ thenメソッドによってオブジェクトを取得する

※ヒント：Promiseオブジェクトについては9章のP.192で解説しています。

5

「Array.prototype.splice()」より引用
引数
start
配列を変更する先頭の位置です。
配列の長さより大きい場合、start は配列の長さに設定されます。この場合、削除される要素はありませんが、このメソッドは追加関数として動作し、提供された item[n] の数だけ要素を追加します。*
https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Global_Objects/Array/splice

① 引数startを配列の長さより大きくしてはいけない

② このメソッドは配列に要素を追加できる

③ 追加関数はアロー関数式で定義する

mission 10-02

達成目標 60 秒

エラー文を見て、その意味を3択から選べ

エラーメッセージを見て、その意味を表すものに丸を付けてください。

```
Uncaught SyntaxError: Unexpected token ')'
```

① 予期せぬ「)」が出現した

② 「)」が抜けている

③ 「)」と話すことは期待されない

1

```
Uncaught TypeError: Cannot set properties of undefined
(setting 'myproperty')
```

① プロパティにundefinedをセットできない

② undefinedのプロパティを読み取れない

③ undefinedのプロパティに値をセットできない

2

```
Uncaught TypeError: datestr.split is not a function
```

① splitプロパティが機能していない

② splitメソッドが定義されていない

③ splitメソッドは関数オブジェクトである

3

```
Uncaught SyntaxError: Identifier 'x' has already been declared
```

① 識別子「x」が重複して宣言されている

② 事前に識別子「x」を宣言しなければならない

③ 識別子「x」を持つメソッドを宣言せよ

mission 10-03

達成目標 60 秒

エラーメッセージを見て、修正指示を書き込め

エラーメッセージを見て、プログラムに修正指示を書き込んでください。

```
let x => 10;
```

Uncaught SyntaxError: Unexpected token '=>'

1

```
let testfunc = () {
    console.log('Test!');
};
```

Uncaught SyntaxError: Unexpected token ')'

2

```
let lst = ['a', 'b', 'c'];
for v of lst{
  console.log(v);
}
```

Uncaught SyntaxError: Unexpected identifier

3

```
let lst = [a, 'b', 'c'];
```

Uncaught ReferenceError: a is not defined

mission 10-04

達成目標 200 秒

呼び出し履歴をたどって エラー原因を探せ

発生しているエラーの原因と思われる部分に下線を引いてください。

```
let splitDateStr = (datestr) => {
  return datestr.split('-');
};

let datestr = 2021 - 1 - 1;
result = splitDateStr(datestr);
```

```
Uncaught TypeError: datestr.split is not a function
  at splitDateStr (……js:2)
  at ……js:6
```

1

```
let getSpan = (stdate, eddate) => {
  let span = eddate.getTime() - stdate.getTime();
  return span;
};

let st = '2020-1-5';
let ed = '2020-2-1';
let span = getSpan(st, ed);
console.log(span);
```

```
Uncaught TypeError: eddate.getTime is not a function
  at getSpan (……js:2)
  at ……js:8
```

2

```
let hasUnderbarClassName = (elem) => {
  return elem.className.indexOf('_') >= 0;
};

let elem = document.querySelector('.chapter_name');
console.log(hasUnderbarClassName(elem.className));
```

```
Uncaught TypeError: Cannot read properties of undefined (reading
'indexOf')
    at hasUnderbarClassName (…….js:2)
    at …….js:6
```

3

```
let hasUnderbarClassName = (elem) => {
  return elem.className.indexOf('_') >= 0;
};

let elem = document.querySelector('.chapter_name');
elem.addEventListener('click', (event) => {
  console.log(hasUnderbarClassName(event));
});
```

```
Uncaught TypeError: Cannot read properties of undefined (reading
'indexOf')
    at hasUnderbarClassName (…….js:2)
    at HTMLParagraphElement.<anonymous> (…….js:7)
```

※hasUnderbarClassName関数は、要素のクラス名にアンダーバーが含まれていればtrueを返します。