復習

確認問題 1

```
//1.足りない型定義を追加しましょう
function isInteger(num) {
  return num >= 0;
}

isInteger(3);

//2.以下2つについてエラーになる原因を考えましょう
isInteger("123");
const numVal: number = isInteger(-10);
```

確認問題 2-1

```
//1.関数を正しく実行できるように型定義を修正しましょう
function BookInfo(book: Book) {
  //省略
}

//使用例
BookInfo({
  title: "TypeScript",
  ISBN: 12345,
  publish: "2022-01-31",
  sale: true,
});

//エラー例
BookInfo({
  title: "JavaScript",
  ISBN: 5678,
});
```

確認問題 2-2

```
//確認問題2-1について以下もエラーなく実行できるようにするためには、型定義をどのように書くとよいでしょうか？
BookInfo({
  title: "JavaScript",
  ISBN: 5678,
});
```

確認問題 3

```
//使用例のように実行できるように関数を修正しましょう
const isInteger: IsIntergerFunc = (num) => num >= 0;

//使用例
console.log(isInteger(-50)); //結果:false
console.log(isInteger(50)); //結果:true
```

確認問題 4

```
//使用例のように実行できるように関数を修正しましょう
function sumScore(arr) {
  return arr
    .filter((num) => num >= 0)
    .reduce((pre, next) => {
      return pre + next;
    }, 0);
}

//使用例
const sum: number = sumScore([100, 98, 200, -300, 500]);
console.log(sum); //結果：898
//エラー例
sumScore(123, 4444);
sumScore([12333, "hogehoge", boolean]);
```

練習問題 1

```
//以下を実行すると結果のような表示なるような関数を実装しましょう。
//税込み前の金額を入力すると税込み後の金額を含めたメッセージを返す関数です。

taxIncluded(100); //結果：合計金額は110円です。
```

練習問題 2

```
//splitが求めているようなtaxIncludedを定義しましょう。
const split = (cost: number, num: number): void => {
  const total: number = taxIncluded(cost);
  //割り勘した結果を返す処理
  const split = total / num;
  console.log(`割り勘金額は${split.toFixed(0)}です`);
};
split(1500, 5); //結果：割り勘金額は330円です。
```

練習問題 3

```
type Fish = ("鯛" | "鯖" | "鰆")[];

let allFish: Fish = [];

allFish.push("鯖");
allFish.push("鯛");
//なぜエラーとなるか考えましょう
//エラーの理由が変わったらコメントアウトしましょう
allFish.push("鯨");

type FishOrNot = string | null;

//型定義をした関数に変えましょう
const checkItem = (item): FishOrNot => {
  //エラーがなくなるように修正しましょう
  return allFish.includes(item) ? "魚です" : 0;
};
//型定義をした関数に変えましょう
const dispCheckResult = (item): void => {
  const result: FishOrNot = checkItem(item);
  console.log(result);
};

dispCheckResult("鯛");
dispCheckResult("鯨");
```

練習問題 4

```
//関数の型を修正しましょう
//また、コードのロジックをコメントで書きましょう
const str = ((fnc: (arg1: number) => string | unknown) , value ) => {
    const s = fnc(value)
    if (typeof s == "string") {
        return s.slice(0, 3);;
    }else{
        return s;
    }
}
console.log(str(val => { val.toString(); }, 10000)); //結果：100
```

練習問題 5

```
//p2においてyのstring型も許容するためにはどのようにコードを修正するとよいでしょうか？
type Point = { x: number; y: number };
type Point2 = Point & { y: number | string };
const p2: Point2 = { x: 20, y: "20px" };
```

練習問題 6

```
//引数はなし、戻り値() => voidのアロー関数
//結果の通りに表示されるように修正しましょう
const timer = (): (() => void) => {
  let counter: number = 0;
  return () => console.log(counter);
};

const timer1: () => number = timer();
const timer2: () => string = timer();

timer1(); //結果：1
timer1(); //結果：2
timer1(); //結果：2

timer2(); //結果：1
timer2(); //結果：2
```