

プログラミング

第6回
条件文

久保田 匠

[準備]授業資料にアクセス

いつもの作業

- 久保田の授業ホームページに資料がアップロードされている。
- まずは「愛教大 数学」と検索してみよう。



The screenshot shows the '授業用ホームページ (久保田)' (Teaching homepage (Kubota)). It includes two tables for '2025年度前期担当科目' (Courses for the first semester of the 2025 academic year) and '2025年度後期担当科目' (Courses for the second semester of the 2025 academic year). In the second semester table, the 3rd and 4th rows are highlighted with red boxes, containing 'プログラミング' (Programming) in both columns. At the bottom left, there is a link to 'その他のコンテンツ' (Other contents) with two small icons. On the bottom right, there is contact information: '数学教育講座 久保田匠 (自然科学棟 521 研究室) Email: skubota [at] auecc.aichi-edu.ac.jp'.

	内容	資料	コード
第1回	いろいろなプログラミング言語 VSCode のインストール	● Prog_01-1	
第2回	Webページを構築する(HTML)	● Prog_02-1	
第3回	Webページの見栄えを整える(CSS)	Prog_03-1 Prog_03-2	
第4回	JavaScriptに触れてみよう	Prog_04-1	
第5回	変数と演算		(なし)
第6回	条件文	● (なし)	
第7回	繰り返し(1)		(なし)
第8回	繰り返し(2)	Prog_08-1	
第9回	繰り返し(3)		(なし)

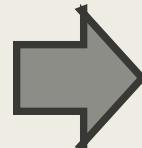
[準備] コードの新規作成①

いつもの作業

- 授業用ホームページからサンプルコードをコピーしよう。

プログラミング

	内容	資料	コード
第1回	いろいろなプログラミング言語 VSCode のインストール	● Prog_01-1	
第2回	Webページを構築する(HTML)	● Prog_02-1	
第3回	Webページの見栄えを整える(CSS)		Prog_03-1 Prog_03-2
第4回	JavaScriptに触れてみよう		Prog_04-1
第5回	変数と演算		(なし)
第6回	条件文	●	(なし)
第7回	繰り返し(1)		(なし)
第8回	繰り返し(2)		Prog_08-1
第9回	繰り返し(3)		(なし)



Prog_04-1

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Prog_04-1</title>
    <!-- 今日はここは使いません。 -->
</head>

<body>
    <!-- ここに今日の授業内容を入力します。 -->
</body>

</html>
```

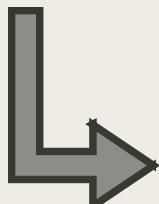
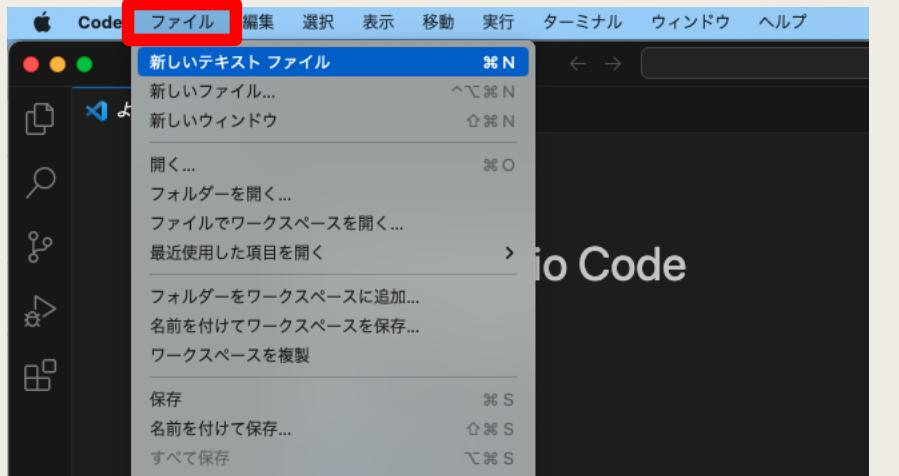
コピー

今日も「Prog_04-1」を選択してください。

[準備] コードの新規作成②

いつもの作業

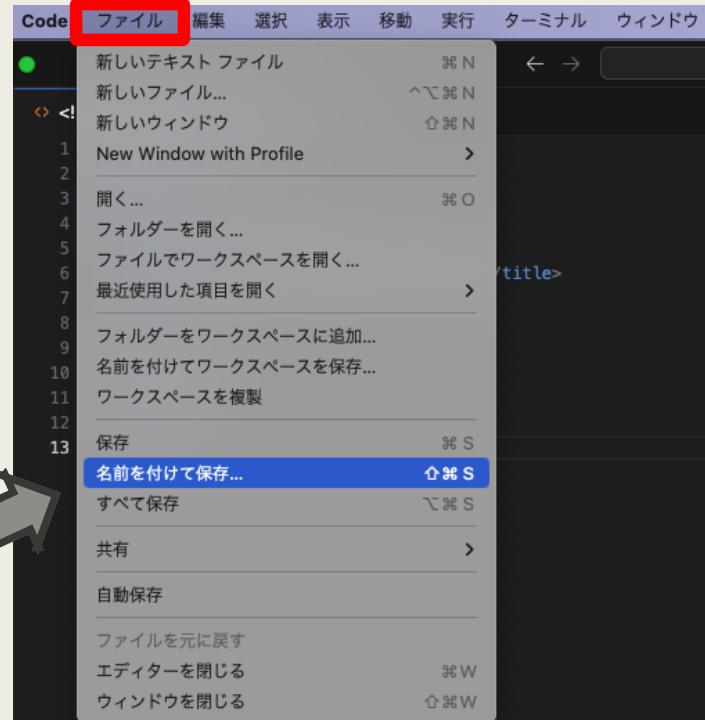
- VSCode を起動し「ファイル」から「新しいテキストファイル」を選択。
- その後、さきほどコピーした文書をペースト（Ctrl + V）して「名前をつけて保存」。



Ctrl + V

A screenshot of an HTML code editor showing the following code:

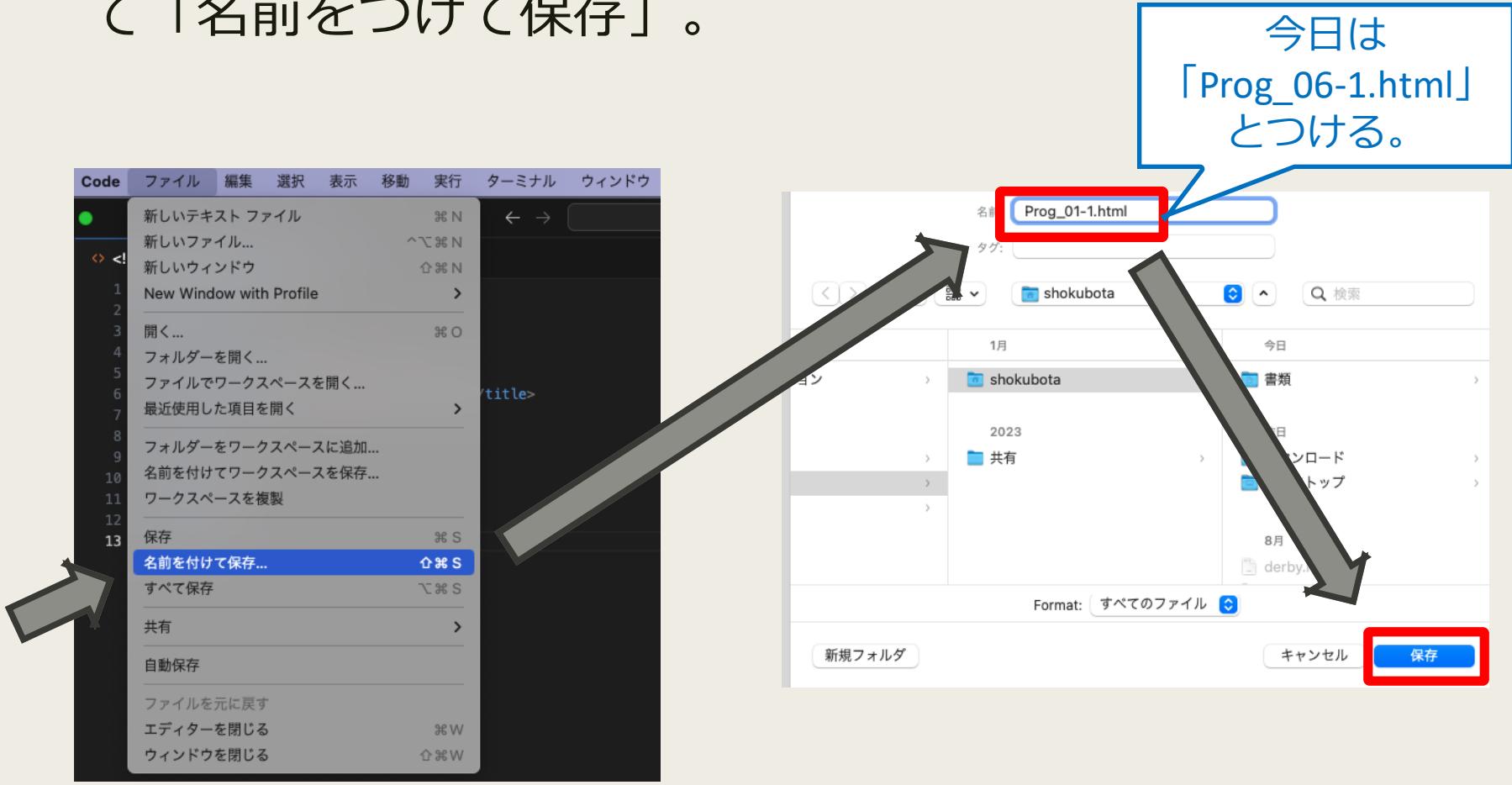
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>ここにページのタイトルを書きます</title>
</head>
<body>
    こんにちは！
</body>
</html>
```



[準備] コードの新規作成②

いつもの作業

- VSCode を起動し「ファイル」から「新しいテキストファイル」を選択。
- その後、さきほどコピーした文書をペースト (Ctrl + V) して「名前をつけて保存」。



[準備]作業環境を整える

いつもの作業

- 保存したhtmlファイルをダブルクリックして開いておく。
- PCの画面をふたつに分け、片方はブラウザ、もう片方はvsCodeを開いておくと便利。

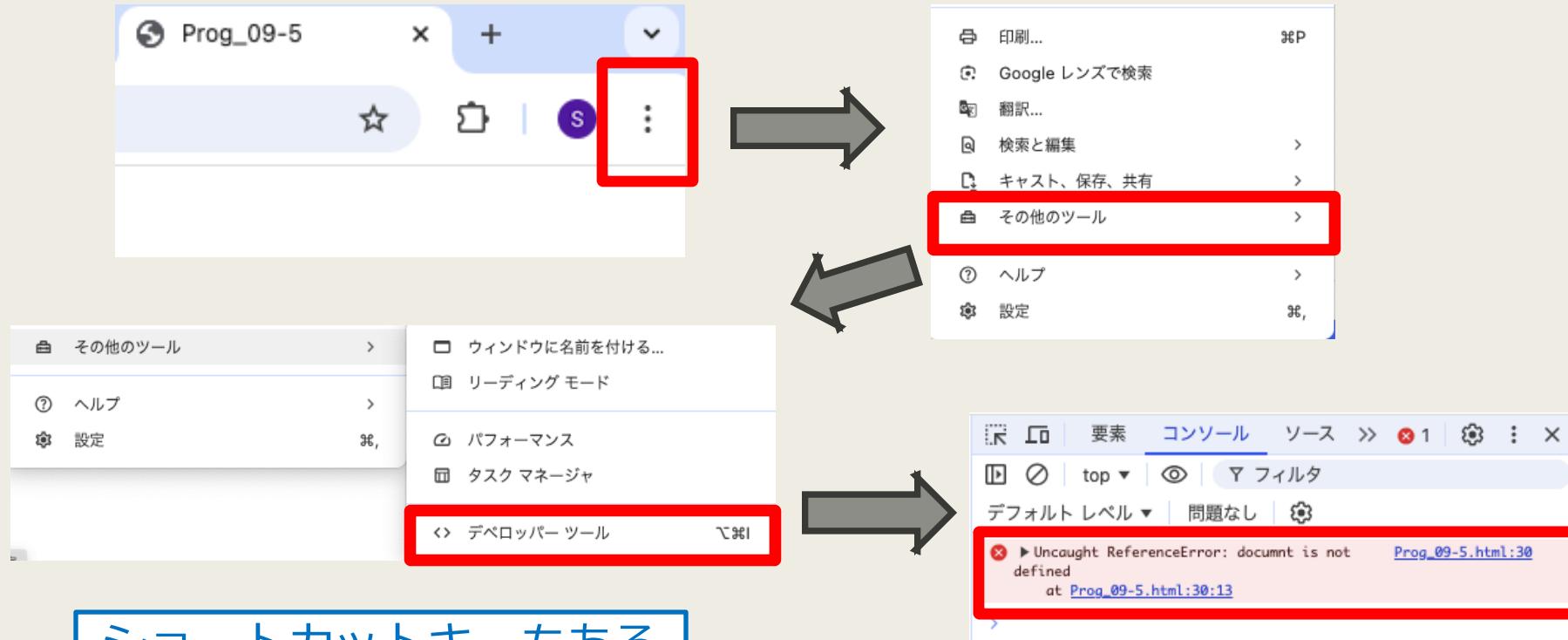


A screenshot of a Mac desktop showing two windows side-by-side. The left window is a web browser displaying the same HTML content as above. The right window is a code editor (VS Code) showing the same HTML code. The code editor has a dark theme and includes status bars at the bottom indicating '行 13, 列 8' and 'UTF-8'.



[復習] デベロッパーツール

- 画面に何も表示されないときや、途中までしか表示されないときはプログラムに間違いがある可能性が高い。
- そのときは「デベロッパーツール」を開き、何行目でエラーが発生しているか見てみよう。



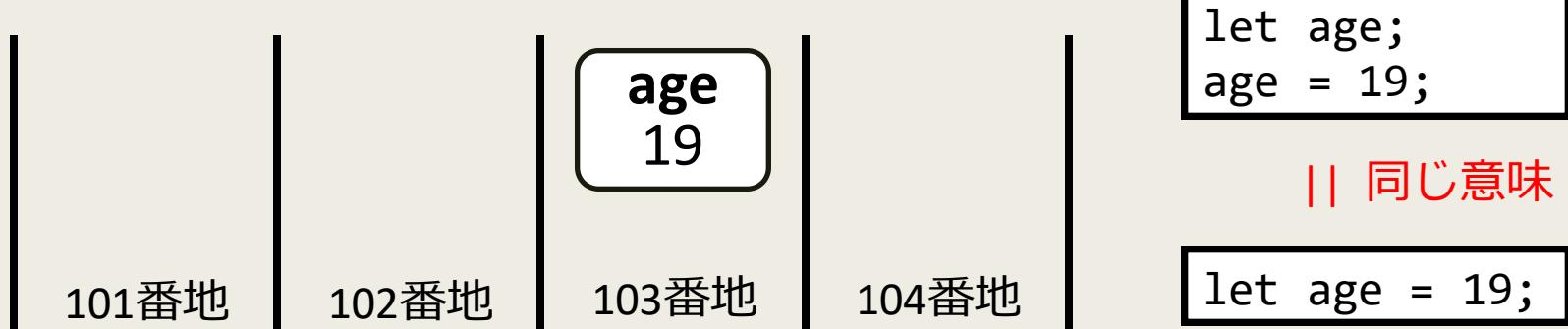
30行目でエラーが発生。
document が未定義と言われている
(スペルミスが発生していた)

[復習]変数

数値だけでなく、文字列、真偽値、オブジェクト、配列も

- プログラミングでは「**値**」を扱う場面が多くある。
- それらの値を一時的に保存する仕組みとして**変数**が用いられる。
- 変数は値を入れる箱のようなもので、ひとつひとつには**変数名**がついている。
- **アドレス**と呼ばれる、箱を置いてある場所を表す数字の列もある（後の学習のために頭の片隅に入れておくとよい）。

例：年齢を管理する変数「age」を用意して 19 を代入



実際のアドレスは 0x7ffee10b2a4c のようにもっと複雑

[復習]いろいろな計算をしてみよう

- プログラミングにおける計算の基本は四則演算と剰余演算。
- 足し算、引き算の演算子はそれぞれ「+」, 「-」。
- 掛け算、割り算の演算子はそれぞれ「*」, 「/」。
- べき乗の演算子は「**」。
- 前回、以下のコードを入力した。

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="UTF-8">
6      <title>Prog_05-2</title>
7  </head>
8
9  <body>
10     <p>
11         <script>
12             let a = 7;
13             let b = 4;
14             document.write("a = 7, b = 4 とする。<br>");
15             document.write("和は" + (a+b) + "<br>");
16             document.write("差は" + (a-b) + "<br>");
17             document.write("積は" + (a*b) + "<br>");
18             document.write("商は" + (a/b) + "<br> である。");
19         </script>
20     </p>
21  </body>
22
23 </html>
```

document.write("和は" + (a+b) + "
");

||

document.write("和は", (a+b), "
");

a = 7, b = 4 とする。
和は11
差は3
積は28
商は1.75
である。

コンマを使って
このように入力しても良い。
console.log() も同様。
alert() ではエラー。

[復習] 剰余演算

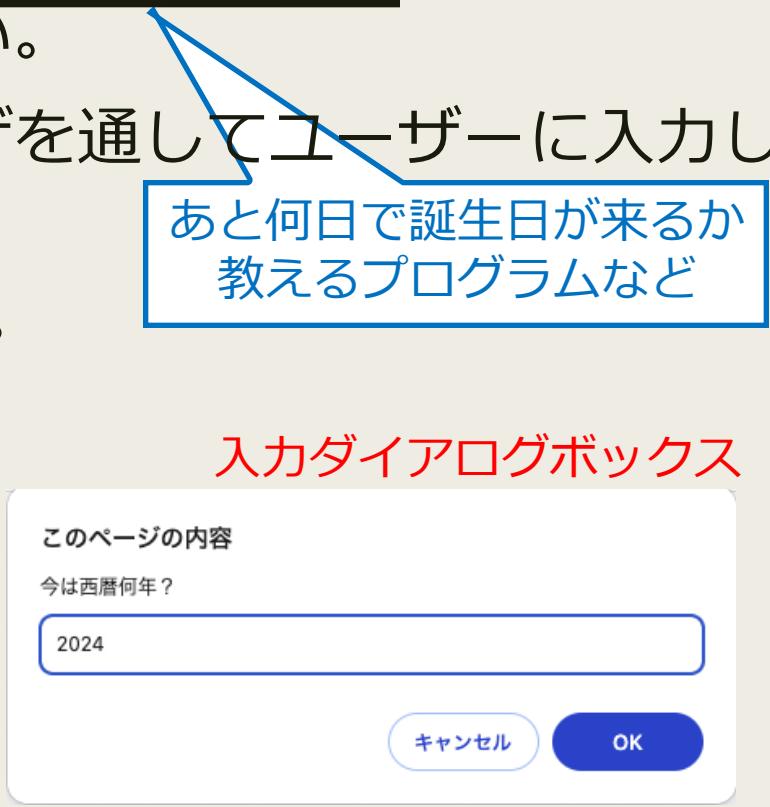
- 剰余演算(%)は余りを求める演算。
 - $13 \% 5 = 3$ ← 13 を 5 でわった余りは **3**
 - $7 \% 2 = 1$ ← 7 を 2 でわった余りは **1**
 - $27 \% 3 = 0$ ← 27 を 3 でわった余りは **0**
- 繰り返し処理を行うときや、繰り返し回数を何かでわった余りによって処理を変えたいときに用いられる。
 - 偶数番目は○○をして、奇数番目は△△をする。
 - 5で割り切れるときに改行(
)をいれる、など。
- 条件分岐（場合分けのようなもの）は今日、繰り返し処理は次回以降扱う予定。

ユーザーから入力を受け取る

- これまでコード内で変数を宣言し、値を代入していた。
- しかし、これでは後から値を変更したいときにそのたびにコードを編集しなければならない。
- 頻繁に値が変わる場合、ブラウザを通してユーザーに入力してもらえば便利である。
- そのための命令に `prompt` がある。
- 次のコードを入力してみよう。

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="UTF-8">
6      <title>Prog_06-2</title>
7  </head>
8
9  <body>
10     <p>
11         <script>
12             let year = prompt("今は西暦何年？");
13             document.write("今は令和" + (year - 2018) + "年です。");
14         </script>
15     </p>
16 </body>
```

// 13行目は次のように入力してもよい。
`document.write("今は令和", year - 2018, "年です。");`



今は令和6年です。

[演習]単位変換プログラム

- オンス(ounce, 記号:oz)とは、重さや液体の単位である。
- 1オンスは 29.5735 ミリリットルである。
- 次の出力結果を参考にし、オンスをミリリットルを変換するプログラムを書いてみよう。

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Prog_06-5</title>
7 </head>
8
9 <body>
10  <p>
11    <script>
12      考えてみよう
13    </script>
14  </p>
15 </body>
16
17 </html>
```



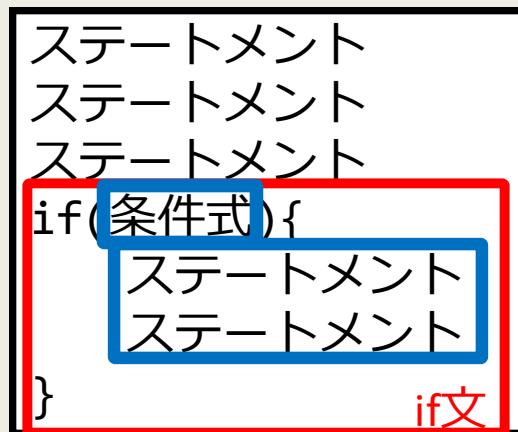
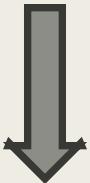
12オンスは354.882ミリリットルです。

- 日本の缶ジュースに 350ml と 500ml が多いのは、アメリカから製缶機を輸入したことと、アメリカで主流の缶サイズが12オンスと16オンスであることが関係している（諸説あり）。

もし〇〇ならば△△を行う

- `if文`は、プログラムが特定の条件を満たすかどうかを確認し、その条件が真(true)であれば処理を実行するという命令。
- 偽(false)であればその処理をスキップする。

通常は上から
ひとつずつ
実行される



- ①条件式が成立したら
- ②{}内のステートメントが
実行される
- ③条件式が整理しなかったら
{}内は実行されない

- 次のプログラムを入力して
みよう。
- 14行目の「`>=`」は「`≥`」と
同じ意味。

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <meta charset="UTF-8">
6    <title>Prog_06-3</title>
7  </head>
8
9  <body>
10 <p>
11   <script>
12     let point = prompt("試験の点数を0から100以下の整数値で入力してください。");
13     document.write("あなたの成績は" + point + "点でした。");
14     if(point >= 60){
15       document.write("単位取得おめでとう！");
16     }
17   </script>
18 </p>
19 </body>
20
21 </html>
```

比較演算子

- if文内の条件式には「true」か「false」かを返すステートメントをいれる。
- そのために用いられる記号（演算子）を **比較演算子** という。

```
ステートメント  
ステートメント  
ステートメント  
if(条件式){  
    ステートメント  
    ステートメント  
}
```

比較演算子	使用例	説明
==	$a == b$	a と bが等しければ true
!=	$a != b$	a と bが等しくなければ true
<	$a < b$	a が bより小さければ true
<=	$a <= b$	a が b 以下ならば true
>	$a > b$	a が bより大きければ true
>=	$a >= b$	a が b 以上ならば true

「=」ではなく
「==」なので注意

そうでないならば□□を行う

- if文のあとに else をつけると条件式が成立しなかった場合の処理を記述することができる。
- 先のプログラムに追記し、試験の点数が60点以上とならなかった場合に「不合格です。また来年頑張ってください。」と表示するプログラムを作ろう。

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="UTF-8">
6      <title>Prog_06-3</title>
7  </head>
8
9  <body>
10     <p>
11         <script>
12             let point = prompt("試験の点数を0から100以下の整数値で入力してください。");
13             document.write("あなたの成績は" + point + "点でした。");
14             if(point >= 60){
15                 document.write("単位取得おめでとう！");
16             } else {
17
18             }
19         </script>
20     </p>
21 </body>
22
23 </html>
```

考えてみよう

```
if(条件式){
    ブロック1
    ブロック2
}
else {
    ブロック3
    ブロック4
}
```

- ①条件式が成立したら
- ②{}内のステートメントが実行される
- ③条件式が成立しなかったら
- ④else {}内のステートメントが実行される

[演習]絶対値を実装する

- ユーザーに実数 x を入力してもらい、その絶対値 $|x|$ を返すプログラムを自作してみよう。
- 次の出力例を参考にすること。



あなたが入力した実数は3.14です。その絶対値は3.14です。

あなたが入力した実数は-1.234です。その絶対値は1.234です。

```
9  <body>
10   <p>
11     <script>
12       let x = prompt("実数を入力してください。その絶対値を言います。");
13       document.write("あなたが入力した実数は" + x + "です。その絶対値は");
14
15
16
17
18
19
20     </script>
21   </p>
22 </body>
```

考えてみよう

「=」と「==」の違い

- 前回も触れたように「=」は代入を行う演算子である。
- 今回扱った「==」は左辺と右辺が等しいなら `true`, そうでなければ `false` を返す比較演算子である。
- なお、if文中の条件式で「`x == 100`」とかくべきところを誤つて「`x = 100`」と入力してもプログラム自体は動いてしまうので注意が必要。
- JavaScript では条件式内の数値が 0 であれば `false` とみなし、それ以外の数値であれば `true` とみなして処理する。

```
let x = 0;  
if(x = 100){  
    ブロック  
} else {  
    ブロック  
}
```

条件式内を処理する



```
let x = 0;  
if(100){  
    ブロック  
} else {  
    ブロック  
}
```

`x`に100を代入し、
100が `true`か `false`かを判断



```
let x = 0;  
if(100){  
    ブロック  
} else {  
    ブロック  
}
```

JavaScriptでは0以外の
数値は `true`と判断

[復習]インデント

- if文などのブロック中のステートメントは、通常のプログラムよりも字下げをするとプログラムの構造が見やすくなる。
- 字下げのことをインデントという。
- インデントの幅は Tabキー1つ分、半角スペース2つ分か4つ分が使われることが多い。
- 幅の取り方は自由に決めて良いが統一させること。
- VSCode では改行に応じてインデントが自動で整えられるが、場合によっては手動で調整が必要になることもある。

```
<body>
<p>
<script>
var point = prompt("試験の点数を0から100以下の整数値で入力してください。");
document.write("あなたの成績は" + point + "点でした。")
if(point >= 60){
document.write("単位取得おめでとう！")
} else {
document.write("不合格です。また来年頑張ってください。")
}
</script>
</p>
</body>
```

インデントがないコード
(構造が分かりにくい)

```
<body>
<p>
<script>
    var point = prompt("試験の点数を0から100以下の整数値で入力してください。");
    document.write("あなたの成績は" + point + "点でした。")
    if(point >= 60{
        document.write("単位取得おめでとう！")
    } else {
        document.write("不合格です。また来年頑張ってください。")
    }
</script>
</p>
</body>
```

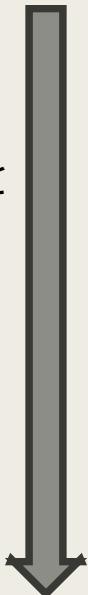
インデントがあるコード
(構造が分かりやすい)

条件を細かく設定する

- else の部分にさらに if を入れることもでき、より細かい条件判断を行うことができる。

上から順に
条件式が
成立するかを
調べていく

```
if(条件式 1 ){  
    //条件式 1 が成立した場合の処理  
} else if(条件式 2 ){  
    //条件式 1 が成立せず条件式 2 が成立した場合の処理  
} else if(条件式 3 ){  
    //条件式 2 が成立せず条件式 3 が成立した場合の処理  
...  
} else {  
    //すべての条件式が成立しなかった場合の処理  
}
```



[演習]else if

- 本資料 p16 のプログラムを修正し、次のように出力するプログラムをかけ。
 - 90点以上なら「あなたの成績は S です。」
 - 80点以上（90点未満）なら「あなたの成績は A です。」
 - 70点以上（80点未満）なら「あなたの成績は B です。」
 - 60点以上（70点未満）なら「あなたの成績は C です。」
 - それ以外なら「あなたの成績は D です。」

p16 のプログラム（コピペ用）

```
<script>
    let point = prompt("試験の点数を0から100以下の整数値で入力してください。");
    document.write("あなたの成績は" + point + "点でした。");
    if(point >= 60){
        document.write("単位取得おめでとう！");
    } else {
        document.write("不合格です。また来年頑張ってください。");
    }
</script>
```

「または」と「かつ」

- 条件式を記述する際、「または」や「かつ」が必要になることがある。
- JavaScript では「または（論理和）」は記号 `||` を用いる。
 - 左右の比較演算式のどちらかが真ならば真を返す。
- JavaScript では「かつ（論理積）」は記号 `&&` を用いる。
 - 左右の比較演算式が両方とも真の場合のみ真を返す。
- 15歳以上65歳未満の年齢を生産年齢期という。次はユーザーに年齢を入力してもらい、その年齢が生産年齢期かそうでないかを表示するプログラムである。入力してみよう。

```
<script>
    let age = prompt("あなたの年齢を入力してください。");
    if((age >= 15) && (age < 65)){
        document.write("あなたは生産年齢期です。");
    } else {
        document.write("あなたは生産年齢期ではありません。");
    }
</script>
```

[演習]教科書を熟読しよう

- 今日の内容は教科書の p84～p103 がベースになっている
- 残った時間で自分でも該当箇所を熟読してみよう。
- 授業で解説していないコードは自分でも入力してみてどのような出力結果になるか確かめてみよう。