STAGE 1-3

変数を使う

データを保存するひとつの手段

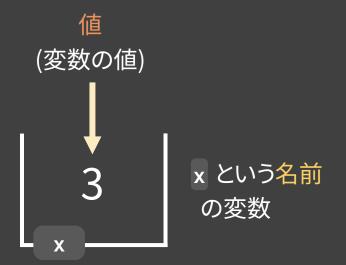
- ► この STAGE の目標
- >「変数」「型」という概念を知る
- > 変数に値を保存したり読み込んだりできる

へん-すう



variable (Vary + able)

プログラムに値を覚えさせるもの、 またはプログラムが覚えている値のこと



全ての変数は、 「名前」と 「値」を持つ

値を入れることができる箱と例えられる

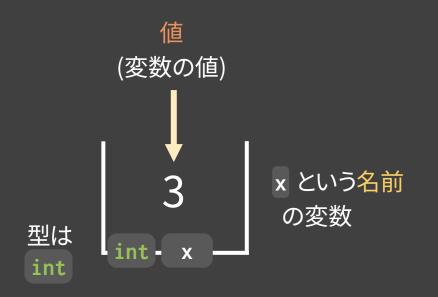
交数 へん-すう



それっぽいコード

xという変数に3を入れる(xという変数の値を3にする)

交数 へん-すう



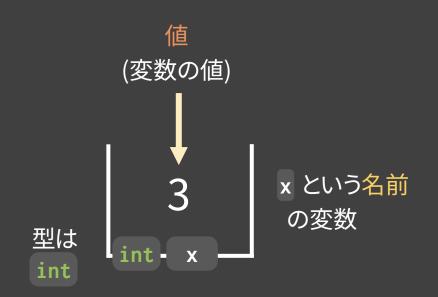
変数には「型」を設定する (C++は必ず指定する)「型」は、変数に入れる値の種類を決める

int は 整数 (±21億 くらいの範囲)

doubleは 小数も整数も入る(だいたい、小数点以下15桁まで正しい)

[注意] 変数の型は途中で変えられない

交数 へん-すう



これは正しいコード

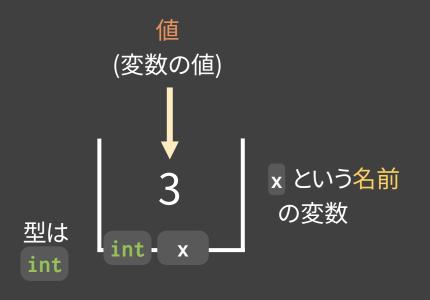
```
1 #include <cstdio>
2
3 int main(void) {
4 int x = 3; // これで実行できる
5 return 0;
6 }
```

xという名前で、int型の変数は 値に3を持っている

変数を使いはじめるときは、「型」「名前」 = 「値」と書く これを、「変数を初期化する」という

変数

へん-すう



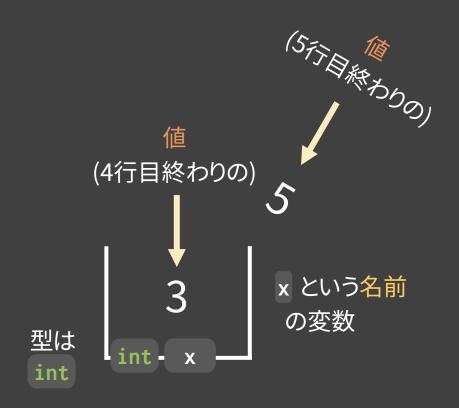


```
1 #include <cstdio>
2
3 int main(void) {
4 int x = 3; // xを初期化した
5 int y; // yを定義した
6 return 0;
7 }
```

```
変数を使いはじめるときは、「型」「名前」 = 「値」と書くこれを、「変数を初期化する」というただし、「値」は書かなくてもOKこの場合「変数を定義する」という
```



へん-すう



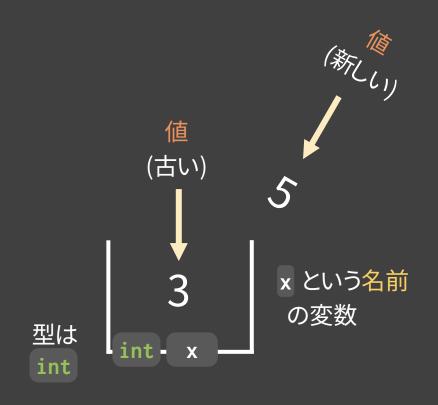
```
1 #include <cstdio>
2
3 int main(void) {
4 int x = 3; // xを初期化した
5 x = 5; // x に 5 を代入
6 return 0;
7 }
```

変数を初期化や定義した後は、その値を変えられる

5行目のように、初期化や定義をした変数には、 「名前」=「値」と書くと値を代入できる これで変数の値を上書きする



へん-すう



¥か∖は Backspace (1文字削除)の左にあるよ

```
#include <cstdio>
   int main(void) {
       int x = 3; // xを初期化した
       printf("%d\u00ebn", x); // xの値
       x = 5; // x に 5 を代入
6
       printf("%d\u00ebn", x); // xの値
       return 0;
9
     printf("%d¥n", 変数の名前);
       で、変数の値を確認しよう
  最初の数字が4行目の初期化後のxの値、
    次の数字が6行目の代入後のxの値
```

变数

へん-すう

#include <cstdio> int main(void) { int x = 3; // xを初期化した printf("%d¥n", x); // xの値 x = 5; // x に 5 を代入 printf("%d¥n", x); // xの値 出力

まだ変数はない

変数 へん-すう

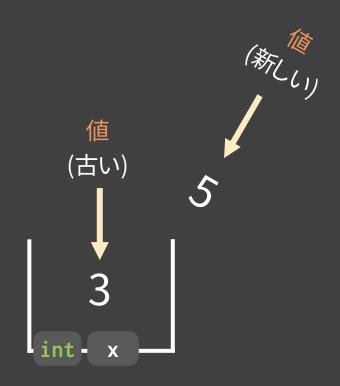


int 型の変数 x を 3 で初期化した

```
#include <cstdio>
       int main(void) {
          int x = 3; // xを初期化した
           printf("%d\u00e4n", x); // xの値
          x = 5; // x に 5 を代入
           printf("%d¥n", x); // xの値
出力
```

变数

へん-すう



変数 x に 5 を代入した

```
#include <cstdio>
      int main(void) {
          int x = 3; // xを初期化した
          printf("%d\n", x); // xの値
          x = 5; // x に 5 を代入
          printf("%d\n", x); // xの値
出力
 3
 5
```

变数

へん-すう



変数 x に 5 を代入した (古い値は上書きされる)

```
#include <cstdio>
       int main(void) {
           int x = 3; // xを初期化した
           printf("%d\forall n", x); // xの値
           x = 5; // x に 5 を代入
           printf("%d\u00e4n", x); // xの値
出力
 3
 5
```

```
printf("%d¥n", y);
yはどっち?

3
int-y

fint-y
```

```
#include <cstdio>
   int main(void) {
       int y = 3; // yを初期化した
5
       int y = 5; // yをまた初期化?
6
       printf("%d¥n", y); // yの値
       return 0;
8
9
```

同名の変数を2度定義や初期化してはいけない

(ただ、後に習う「スコープ」に依存する)



変数を使いはじめるには

 定義
 int x;
 「型」「名前」

 初期化
 int x = 0;
 「型」「名前」=「値」

変数の値を使う (参照する) には 変数名をそのままコードに入れる

printf("%d\u00e4n", x);

変数の値を書き換えるには

代入 x = 1;

「名前」=「値」

printf() の使い方は 次節で説明します

- ▶ この STAGE の目標
- >「変数」「型」という概念を知る
- > 変数に値を保存したり読み込んだりできる

できるようになったこと

変数の定義や初期化、代入ができる 変数の型がわかる