

質問 1 次のコードの表示がうまくいきません。どうしてですか？

```
int a=5, b=7;
int add=a+b;
System.out.println(a+"b"="add);
```

回答

```
System.out.println(...);
```

の中の、各リテラルや変数の間に、文字列連結のための演算子 + が必要です。

```
System.out.println(a + "+" + b + "=" + add);
```

とするとよいです。

演算子 + には、2つの機能があることを思い出しましょう。
加算と文字列連結（文字列リテラルとその他のリテラルを連結します）
です。

これらの機能は演算子のパラメータにより切り替わるということでしたね。
文字列連結 ← 片方 または 両方のパラメータが String 型（文字列リテラルを含む）
加算 ← それ以外

上の例のように出力したい文字列を連結したい場合には、
System.out.println(a + "+" + b + "=" + add);
のようにパラメータの一方に文字列がくるようにしてやれば文字列連結になります*。

※これ以外の場合でも文字列連結になります。これは演算の優先順位に関係しますので以降の回で詳しく解説します。

質問 2 次のコードの表示がうまくいきません。どうしてですか？

```
int num=1234;
double gas=10.5;
System.out.println("車のナンバーは+"num"+です。");
System.out.println("車のガソリン量は+"gas"+です。");
```

回答

```
System.out.println(...);
```

の中で、文字列連結のための演算子 + が文字列リテラルの " "の中に入っています。

```
System.out.println("車のナンバーは" + num + "です。");
System.out.println("車のガソリン量は" + gas + "です。");
```

とするとよいです。

文字列リテラルと変数の間には文字列連結のための演算子 + が必要です。

質問3 次のコードがコンパイルできません。どうしてですか？

```
int a=5, b=7;
add=a+b;
int add;
System.out.println(a + "+" + b + "=" + add);
```

回答

変数 `add` が宣言されるよりも前に代入で使用されています。変数は、宣言された直後から使用できます。

```
int a=5, b=7;
int add;
add=a+b;
System.out.println(a + "+" + b + "=" + add);
```

とするとよいです。

質問4 次のコードは変数 `c` の初期化をしていませんが大丈夫でしょうか？

```
int a=1, b=2, c;
c=a+b;
System.out.println(c);
```

回答

変数は、宣言された直後から利用可能になります。宣言された後、その変数の値を読む（参照する）コードが実行されるまでに、その変数に何も値が代入されない場合、コンパイルエラーになります。

上のコードでは、

```
int a=1, b=2, c;
c=a+b;
System.out.println(c);
```

赤のコードで、変数 `c` に値が入っており、変数 `c` の値を出力するコードを実行するときにはすでに値が代入されています。したがって、コンパイルエラーとはなりません。

しかしながら、変数を宣言したら `0` などの値をすぐに入れておく習慣をつけておくといいでしょう。