

# while文

▪ while文とは...

条件式が真になるまで繰り返す。以下は書き方の例。

```
void roop() {  
    while(true) {  
        System.out.println("Hello, world!!");  
    }  
}
```



**注意: 実行しないでください**

# while文.2

- while文の条件式...

条件式の箇所にtrueと書くと、while文全体を何度も繰り返す。  
→なので、先ほどの例を実行すると、コンソール画面に"""Hello,  
world!!"""が大量に出力される。

- while文を終了するためには...

条件文をちゃんとした論理式にしたり、if文の条件分岐でbreak;  
の結果を返す。

# while文.3

- ・以下は1から入力された値までの和を求める例題である。

```
package lesson01;

import javax.swing.JOptionPane;

public class Summation {
    public static void main(String args[]){
        new Summation().start();
    }

    void start(){
        String input = JOptionPane.showInputDialog("1からnまでの総和を求めます。整数を入力して下さい");
        int n = Integer.parseInt(input);
        int sum = 0;

        while(n != 0){
            sum += n;
            n -= 1;
        }

        JOptionPane.showMessageDialog(null, "1からnまでの総和は" + sum);
    }
}
```

# while文.4

- 例題Summationの別の書き方

```
1 package lesson01;
2
3 import javax.swing.JOptionPane;
4
5 public class Summation2 {
6     public static void main(String args[]){
7         new Summation2().start();
8     }
9
10    void start(){
11        String input = JOptionPane.showInputDialog("1からnまでの和を求めます。整数を入力して下さい");
12        int n = Integer.parseInt(input);
13        int sum = 0;
14        while(true){
15            if(n != 0){
16                sum += n;
17                n -= 1;
18            }
19            else{
20                break;
21            }
22        }
23        JOptionPane.showMessageDialog(null, "1からnまでの総和は" + sum);
24    }
25 }
```

break;でwhile文を終わらせる

# 色々なメソッドの型

- ・この講座で扱うメソッドの型の種類...

だいたい void String int double booleanなどを扱う。

今回はvoid型のメソッドについて説明する。

# void型メソッド

- voidとは...

voidは虚とか、空とかいう意味。(by Google先生)

その名の通り、返す値がない。startメソッドもvoid型。

- プログラム内でのメソッドの定義

次のページに例題を示す。これを参考にしてください。

# void型メソッド.2

- void型メソッドを使って、変数を書き換える例題

```
package lesson01;
```

```
import javax.swing.JOptionPane;
```

```
public class Void {  
    public static void main(String args[]) {  
        new Void().start();  
    }  
}
```

```
int n;
```

```
void start() {  
    String input = JOptionPane.showInputDialog("整数を入力して下さい");  
    inputNumber(input); //クラス変数nに値を入れるメソッド  
    JOptionPane.showMessageDialog(null, "入力された値の2乗は" + n * n);  
}
```

```
void inputNumber(String number) {  
    n = Integer.parseInt(number);  
}
```

```
}
```

このクラス内で新しく定義しなくても使えるようになる！

メソッド名(受け渡す変数);  
でメソッドを呼び出す

メソッドの型 メソッド名(変数の型 変数名)  
で起動

## void型メソッド.3(補足)

- ・メソッドの呼び出し→起動について...

例題のように、メソッドを呼び出したいときには

**メソッドの名前(受け渡したい変数);**

と書く。

起動するときには

**メソッドの型 名前(受け渡される変数の型 変数名)**

と書く。このとき、受け渡される変数の名前を変えることができる。

先程の例題で言うと、inputと受け渡しているのに対してnumberという名前で受け取っている。

# 演習.4

while文を用いて、1から入力された値までの積を求め、ダイアログ表示するプログラムを作成せよ。

package名 : lesson01

class名 : Factorial

# 答え

```
1 package lesson01;
2
3 import javax.swing.JOptionPane;
4
5 public class Factorial {
6     public static void main(String args[]){
7         new Factorial().start();
8     }
9
10    void start(){
11        String input = JOptionPane.showInputDialog("1からnまでの階乗を求めます。整数を入力して下さい");
12        int n = Integer.parseInt(input);
13        int fact = 1;
14
15        while(n != 0){
16            fact *= n;
17            n -= 1;
18        }
19
20        JOptionPane.showMessageDialog(null, "1からnまでの階乗は" + fact);
21    }
22 }
```