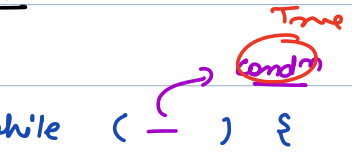
# while loops

```
lop ("Hello")
lop ("Hello")
lop ("Hello")
lop ("Hello")
lop ("Hello")
```

doops




3

```
int cnt=1;
while (cnt <=5) {
    lop("Hello");
    cnt= cnt+1 | cnt++=1; | cnt++;
}
```

## Syntax

1. initialization

```
while ( condn ) {
    // work
    updation
}
```

```
int cnt=1;
while (cnt <=5) {
    lop("Hello");
    cnt= cnt+1 | cnt++=1; | cnt++;
}
```

Ques Given a no., print its digits.

N = 6581 o/p → 1, 3, 5, 6

N = 123 o/p 3, 2, 1

Extract last digit
↳ n%10

N = 6581

↓ /10 o/p → 1

659 %10 = 3

↓ /10

65 %10 = 5

↓ /10

6

Incorrect ↴

1) n = Scn.nextInt();
2) while(n>=0) {
3) cop(n%10); →
4) n = n/10
}

n = 6581 65 6 0

o/p :

3

5

6

0

```
if (n==0) { print 0 } 3
else {
    n = scn.nextInt();
    while(n > 0) {
        sop(n%10);
        n = n/10
    }
}
```

Ques) Given an Integer (n), print sum of its digits. $n \geq 0$

N = 6531 o/p → 15

N = 123 o/p → 6

```
n = scn.nextInt();
int ans = 0;
while(n > 0) {
    ans += (n%10);
    n = n/10
}
sop(ans);
```

**Ques** Given a number N, Print reverse the no.

→ N = 6531, O/P → 1356

→ N = 123, O/P → 321

rev = rev * 10 + extracted

N = 6531, Extracted

6531 → 653 → 65 → 6 → 0

1 → 3 → 5 → 6

rev = 0 * 10 + 3 => 30 + 5 => 1350 + 6 => 1356

rev

6531 → 653 → 65 → 6 → 0

13

rev = rev * 10 + rem

rev = 1 → 1 * 10 + 3 → 13 * 10 + 5 → 135 * 10 + 6 → 1356.

6531 → 653 → 65 → 6 → 0

1 → 3 → 5 → 6

n = 123

```
n = sc.nextInt();   123  12  1  0
int rev = 0;  →  ~~3~~  ~~32~~  321
while(n > 0) {
    rev = (rev * 10) + (n % 10);
                320 +   1
    n = n / 10
}
sop(ans);
```

```
n = sc.nextInt();
int rev = 0;
while(n > 0) {
    int dig = n % 10;
    rev = (rev * 10) + dig
    n = n / 10
}
sop(ans);
```

https://www.scaler.com/topics/java/online-java-compiler/?snippet_slug=7d5e6a9cba209df4ed77