



ድግግማዊ ቃላት (Loops)

5.1 ምዕላደ - ቃላት

እማርኛ ቃል	እንግሊዘኛ ቃል
ድግግም	repetition
ዙር/ዙሪያ	iteration
ዙር ቆጣሪ	loop counter
ተፈታኝ ቃል	test condition

5.2 የድግግማዊ ቃላት ዓላማ

በልዩ ልዩ ሁኔታዎች መደጋገም የሚጠይቁ ችግሮች ሥፍር ቀጥሮ የላቸውም። ድግግማዊ ቃላት ፣ መመሪያዎች ከአንድ ጊዜ በላይ ተከታትለው በሥራ ላይ እንዲውሉ ያስችሉናል። ለምሳሌ ከ1-5 ያሉትን ቁጥሮች በተረታ ማባዛት ብንፈልግ ፣ ያለአስደጋጫ ቃላት ይኸን ይመስላል።

```

ድግግሞሽ
int result = 1 ;

result = result * 1 ;
result = result * 2 ;
result = result * 3 ;
result = result * 4 ;
result = result * 5 ;

```

ምንም እንኳን ሌሎች ችግሮች ቢነሩም ፣ ለጊዜው ድግግሞሽ ላይ እናተኩር። በግልጽ እንደምናየው ፣ መፍትሔው እምስት መመሪያ ቃል ይዟል። ምናልባት ይኸ ተቀባይነት ይኖረው ይሆናል። ነገር ግን ፣ ችግሩ ከ1-5 ማባዛት መሆኑ ቀርቶ ከ1-100 ብናደርገው ፣ የመመሪያዎቹ ቃል ብዛት በዛው መጠን ይጨምራል። ገደቡን ወደላይ እየሰቀልን ስንመጣ ፣ የመመሪያዎች ቁጥር ይንፈራል። እንደዚህ ዓይነቱ ፕሮግራም አጻጻፍ ፣ የኮድ መንዛዛት ከማስከተሉም በላይ አንብቦ ለመረዳት እንዲሁም ግድፈትን ለማረም ወይም መሻሻልን ለማምጣት እንቅፋት ይሆናል። ድግግማዊ ቃላትን በመጠቀም የተሻለ መፍትሔ ካጠረና ከነጠረ ኮድ ጋር መጻፍ ይቻላል። ለምሳሌ ፣

```

የfor ቃል

int result = 1 ;
int until = 1000 ;

for (int n=1; n < until; n++)
    result = result * n ;

```



ይህ የተሻለና ቅልብጭ ያለ መፍትሔ ነው። ለማንበብ ፣ ለማሻሻል ፣ ወይም ግድፈት ለማረም ቀላልና አመቺ ነው። ስለዚህ ድግግሞሽ ቃላት ለፕሮግራም ጸሐፊዎች ብርቱ ኃይል ይለግሳሉ።

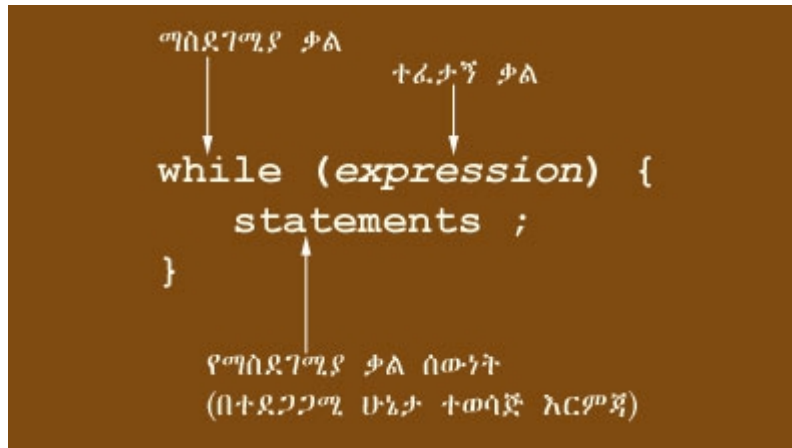
5.3 የድግግሞሽ ቃል ዓይነታት

በጃቫ ቋንቋ አራት የድግግሞሽ ቃላት አሉ። በዚህ ምዕራፍ ሦስቱን ብቻ እንመለከታለን። ሦስቱም መሠረታዊ ተግባራቸው አንድ አይነት በሆነም ፣ ግልጽና የማያሻማ ልዩነት አላቸው። በመሆኑም አንደኛው ከሌላኛው ለአንድ ችግር ያመቸ መፍትሔ ሊሆን ይችላል።

- የwhile ድግግሞሽ ቃል
- የfor ቃልና
- የdo while ቃል ናቸው።

5.4 የwhile ድግግሞሽ ቃል

የwhile ቃል አገባብ



የዚህ ድግግሞሽ ቃል የሥራ ቅደም-ተከተል እንዲህ ነው።

- ሂደቱ እስከ ሲደርስ ፣ ተፈታኝ ቃል ይገመገማል። ውጤቱ አዎ ከሆነ ፣ ሂደቱ ወደ ሰውነቱ ያመራል።
- በውስጡ ያሉት መመሪያዎች በሥራ ላይ ይውላሉ።
- የውስጡ ሥራ እንደተጠናቀቀ ፣ ሂደቱ ወደ ተፈታኝ ቃል ወይም ወደ ድግግሞሽ ቃል ራሱ ይመለሳል።
- ተፈታኝ ቃል እንደገና ይገመገማል። ውጤቱ አዎ ከሆነ በተፈ ቁጥር 2 እና 3 የተጠቀሱት ይከናወናሉ። ነገር ግን ውጤቱ አይደለም ከሆነ ፣ የድግግሞሽ ቃል ዙር ይሰበርና ወደ ተከታዩ መመሪያ ሂደቱ ያመራል።

ምሳሌ ፣ ይህ መመሪያ የተሰጠውን የቁጥር አባላት አንድ በአንድ በተረታ ያወጣና ይጽፋቸዋል። እንደዚህ አይነት ዘዴ ከቁጥር ፳ የተያያዙ ችግሮችን መፍትሄ ለመስጠት ይደግፋል።

የwhile ቃል

```
int number = 314159 ;

while (number > 0) {
    System.out.println(number % 10) ;
    number /= 10 ;
}
```

ማብራሪያ፦

- በፊታኛ ቃል የnumber ዋጋ ከ0 በላይ እስከሆነ ድረስ ፣ የገምገማው ውጤት አዎ ይሆናል። በመሆኑም ቃሉ ራሱን እየመለለሰ ይደጋግማል። ያ ማለት ፣ በውስጡ ያሉት መመሪያዎች በተደጋጋሚ በሥራ ላይ ይውላሉ።
- አስደጋግሞ ቃል ራሱን የሚገታው የnumberን ተውላጠ-ቃል በማደስና ዋጋው ዜሮ ወይም ከዛ በታች ሲሆን ነው።

የሚከተለው ፕሮግራም የwhile የደገገም ቃል አሠራር እንዲሁም ጠቀሜታ ለማሳየት ይሞክራል።

Conversion.java

```
public class Conversion {

    /* Convert decimal to binary and returns the result */
    String toBin(int n) {
        String bin = "";

        while (n > 0) {
            bin = "" + n % 2 + bin ;
            n /= 2 ;
        }
        return bin ;
    }

    /* Convert decimal to hexadecimal and returns the result */
    String toHex(int n) {
        String hexDigits = "0123456789ABCDEF" ;

        String hex = "";

        while (n > 0) {
            char c = hexDigits.charAt(n%16) ;

            hex = "" + c + hex ;
            n /= 16 ;
        }
        return hex ;
    }

    /** An entry point for program execution */
    public static void main(String[] args) {
        Conversion con = new Conversion() ;
    }
}
```

```

int decimal = 1024 ;
System.out.println("in decimal " + decimal) ;
System.out.println("in binary " + con.toBin(decimal)) ;
System.out.println("in hexadeciaml " + con.toHex(decimal)) ;

    }
}

```

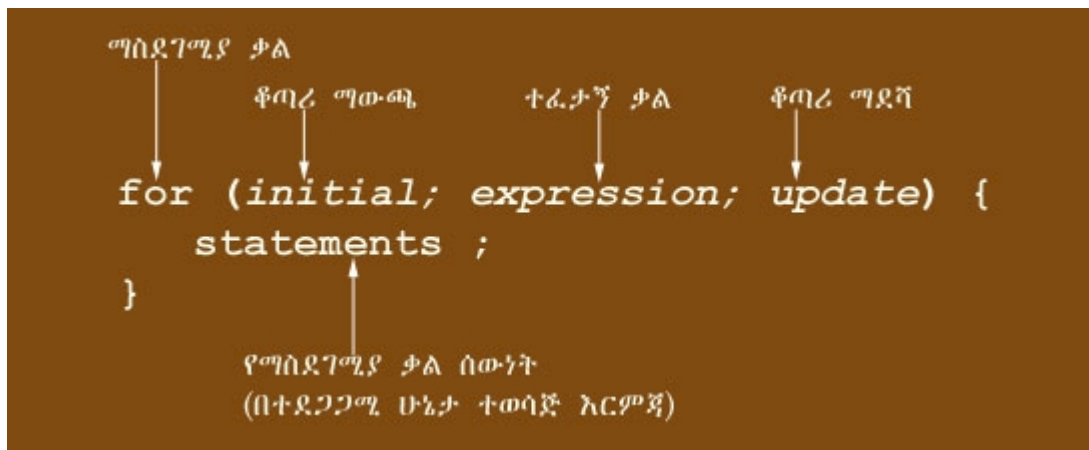
[Download: Conversion.java](#)

5.5 የfor ድግግሞሃ ቃል

የwhile ድግግሞሃ ቃል ፣ የተሰጠውን የሚቆጣጠረው ወይም ሥራው መደገም ወይም መታገት እንዳለበት ውሳኔ አደራረሱ ብዙ ጊዜ ከሰውነቱ ተግባራት ጋራ የተያያዘ ነው። አስቀድሞ ያየነው ምሳሌ ይኸን ያመለክታል።

የfor ድግግሞሃ ቃል ከwhile ግን ለየት ይላል። የድግግሞሃችን ህልውና የሚቆጣጠረው የራሱ የሆነ ክፍል አለው። በተጨማሪ ፣ የድግግሞሃ መጠናቸው በግልጽ ለታወቀ ችግሮች ላሠራር አመቺና ተመራጭ ነው።

የfor ቃል አገገብ



ምሳሌ፦ ይህ ድግግሞሃ ቃል ከ1-1024 ያሉትን ቁጥሮች በተርታ ይደምራል። ውጤታቸውን በsum ተውላጠ-ቃል ይጠብቃል።

```

የfor ቃል

int sum = 0 ;

for (int i=1; i < 1024; i++) {
    sum = sum + i ;
}

```

ይህ ምሳሌ እንደሚያሳየው ፣ ቃሉ ድግግሞሃን የሚቆጣጠረው አናቱ ላይ ባለው ክፍል ነው። የድግግሞሃ ቆጣሪ በመጀመሪያው ክፍል ታውጇል ፤ ቀጥሎ ተፈታኝ ቃል ገብቷል ፤ ከዛ የቆጣሪው ማደሻ ቃል ተጨምሯል። የድግግሞሃ ሰውነት በጥምዘ ቅንፍ የተከበበው ነው።

የሂደቱ ቅደም-ተከተል እንዲህ ነው።

1. የመነሻ ክፍሉ ድግግሞሽ ከመጀመሩ በፊት ይታያል። አንድ ጊዜ ብቻ። በዚህ ምሳሌ ፣ የድግግሞሽ ቆጣሪ የወጣውና መነሻ ዕሴቱ የተሰየመው እዛ ነው።
2. ድግግሞሽ ቃል በሥራ ላይ መዋል እንዳለበት ወይም እንደሌለበት ለመወሰን ተፈታኝ ቃል ይገመገማል። ውጤቱ አዎ ከሆነ ድግግሞሽ ይቀጥላል። ካልሆነ ግን ድግግሞሽ ይቋረጣል።
3. የተፈታኝ ቃል የግምገማ ውጤት አዎ ከሆነ የድግግሞሽ ሰውነት ውስት ያሉት መመሪያዎች በሥራ ላይ ይውላሉ።
4. ዙሩ እንዳለቀ ፣ ሂደቱ ወደ ቆጣሪ ማደሻ ክፍል ይዘዋወርና እዛ ውስጥ የተጠቀሰው ተግባር ይፈጸማል።
5. የተፈታኝ ቃል ቀጥሎ ይገመገምና እንደ ውጤቱ ተከታዩ እርምጃ ይወሰዳል።

ሌላ ምሳሌ፡

```
የfor ቃል

for (int i=1; i < 100; i++) {
    if (i % 11 == 0)
        System.out.println("Divisible by 11: " + i) ;
}

አንደገና ሌላ ምሳሌ፡-

String text = "AakjjEPW;akQ" ;

for (int i=0; i < text.length(); i++) {
    if (text.charAt(i) >= 'a' && text.charAt(i) <= 'z')
        System.out.println("Lowercase: " + text.charAt(i)) ;
    else if (text.charAt(i) >= 'A' && text.charAt(i) <= 'Z')
        System.out.println("Uppercase: " + text.charAt(i)) ;
    else
        System.out.println("Non-alphabet: " + text.charAt(i)) ;
}
```

ይኸንን ክፍል የforን ድግግሞሽ ቃል በሥራ በሚያሳይ ፕሮግራም እንዘጋለን። አንባቢው እንደሚረዳው ፣ ስለፕሮግራሙ የቀረበ ማብራሪያ የለም። አንደኛው ምክንያት አንባቢው አንደ መለስተኛ ፈተና እንዲመረምረው ነው።

```
Drawing.java

public class Drawing {

    /* Using '*', it draws a triangle */
    void triangle(int n) {
        for (int i=0; i < n; i++) {
            for (int j=0; j < i; j++)
                System.out.print('*') ;
            System.out.println() ;
        }
    }
}
```

```

/* Using '*', it draws a rectangle */
void rectangle(int n) {
    for (int i=0; i < n; i++) {
        for (int j=0; j < n; j++)
            System.out.print('*') ;
        System.out.println() ;
    }
}

/** An entry point for program execution */
public static void main(String[] args) {
    Drawing drawing = new Drawing() ;
    int size = 10 ;
    drawing.triangle(size) ;
}
}

```

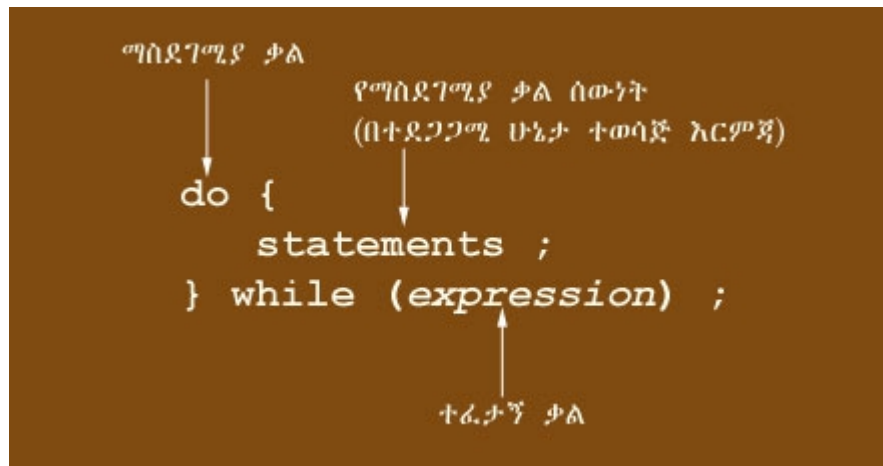
[Download: Drawing.java](#)

5.6 የdo while ድግግሞሃ ቃል

ይህ ድግግሞሃ ቃል ከwhile ጋዙ በሁሉም ነገር ተሰማምቶ በአንድ ነገር ብቻ ይለያል። የተፈታኑ ቃል የሚገመገመው ሁልጊዜ ከድግግሞሃ ቃል መጨረሻ ነው። በሌላ አነጋገር ተፈታኑ ቃል እንደ while ራሱ ላይ ሳይሆን ግርጌው ላይ ነው ያለው። በመሆኑም ምንም ፈተና ሳይደረግ የመጀመሪያው ዙር በሥራ ላይ ይውላል።

የdo while ቃል የሚሰጠው ተጨማሪ ጠቀሜታ ውስን ነው። አንዳንድ ጊዜ ላጠቃቀም ተመራጭነት ይኖረው እንደሆነ እንጂ ፣ አሁን የሚሠራውን በመሉ የwhile ቃል መሥራት ይችላል።

የdo while ቃል አገባብ



ምሳሌ፦

የdo while ቃል

```
String s = "" ;

do {
    s = s + "a" ;
    System.out.println(s) ;
} while (s.length() < 1) ;
```

የመጀመሪያው ዙር ነፃ ነው። ያለምንም ፈተና በድግግሙ ሰውነት ውስጥ ያሉት መመሪያዎች በሥራ ላይ ይውላሉ። ከአንድ ዙር በኋላ ፣ ተፈታኝ ቃል ይገመገማል። ውጤቱ አዎ ከሆነ ድግግሙ ይቀጥላል። አለባ ፣ ይቋረጥና ወደ ተከታዩ ሂደት ያመራል።

መዋቅሩ ላይ በመመርኮዝ ወደ while ቃል ብንቀይረው ይኸን ይመስላል።

የwhile ቃል

```
String s = "" ;

while( s.length() < 1) {
    s = s + "a" ;
    System.out.println(s) ;
}
```

የፍ መጠን ዘሮ ስለሆነ ተፈታኝ ቃል ሲገመገም ውጤቱ አይደለም ይመጣል። ስለዚህ ፣ ድግግሙ ምንም አይነት ዙሪያ አይኖረውም። በውስጡ ያሉት መመሪያዎች በሥራ ላይ አይውሉም።

5.7 የbreak እና የcontinue ቃላት

አንዳንድ ጊዜ የአስደጋጫ ቃላትን ሂደት ወይም ዙሪያዎች ማቋረጥ አሥፈላጊ ይሆናል። ምንም እንኳን የድግግሞሹን ሂደት ይቆጣጠር ዘንድ ተፈታኝ ቃል ቢኖርም ፣ አንዳንድ ጊዜ ሂደቱ የተፈታኝ ቃልን ግምገማ ላያስጠብቅ ይችላል። በእንደዚህ አይነት ሁኔታ የbreak ቃል የድግግሙን ቃል ሰብሮ ለመውጣት የሚያስችል ነው።

አንድ ፕሮግራም ጸሐፊ የግድ የbreakን ቃል መጠቀም አይኖርበትም። ይሁን እንጂ ፣ አንዳንድ ሁኔታዎች ላይ መጠቀሙ አመቺ ሆኖ ይገኛል። ለዛ መዳዳት አያስፈልግም።

ምሳሌ፦

የfor ቃል

```
int n = 1 ;

for ( ; ; ) {
    if ( n%20 == 0)
        break ;
    else
        n = n + 1 ;
}
```



ይህ አስፈጻሚ ቃል ዙፋን የሚቆጣጠረው ክፍሉ ባዶ ነው። በጃቫ ሕግ መሠረት የዙፋን መጠን መጨረሻ የለውም። ለዘላቅም ሲዞር ይኖራል። ደግነቱ ፣ በውስጡ ቃል የተጠቀሰው ሁኔታ ሲግላል። ደግግሙን ሰብሮ ይወጣል። በዚህ መንገድ የአስፈጻሚው ቃል ዕድሜ ተገደበ።

የcontinue ቃል ለየት ያለ ጥቅም አለው። የbreak ቃል ደግግሙን ሙሉ በሙሉ ሰብሮ ለመውጣት ሲሆን ፣ የcontinue ቃል ግን ዙፋኑን አቋርጦ ወደ አስፈጻሚው ቃል ራሱ ሂደቱን ለማምራት ነው።

ምሳሌ፦

```

while ቃል

int n = 1 ;

while (n < 16) {
    if ( n%2 == 1)
        continue ;

    // do something here
    n++ ;
}

```

የn ዕሴት በሁለት ተካፍሎ ቀሪው አንድ ከሆነ ዙፋኑ ይቋረጥና ወደ ተፈታኙ ቃል ያመራል። አለዛ ፣ የቀሩት መመሪያዎች በሥራ ላይ ውለው ፣ እንደተለመደው ሂደቱ ወደ ተፈታኙ ቃል ያመራል።

```

while ቃል

int n = 1 ;

while (n < 16) {
    if ( n%2 == 1)
        continue ;

    // do something here
    n++ ;
}

```