

آموزش زبان برنامه نویسی جاوا

کلاس Math

جلسه سیزدهم

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



در جلسات قبل با کتابخانه ها ، پکیج ها و کلاس های آماده جاوا و چگونگی استفاده از آن ها آشنا شدیم .
یاد گرفتیم که جاوا سری کلاس ها و دستورات آماده را برای ما فراهم کرده که راحت تر کد بزنیم.
در این جلسه قصد داریم یکی دیگر از این کلاس های آماده را بررسی کنیم.

ما گاهی در برنامه نویسی نیاز عملیات محاسبات ریاضی نظیر توان ، قدر مطلق ، عملیات مثلثاتی نظیر سینوس ، کسینوس ،
لگاریتم و..... داریم.خب باید چکار کنیم؟!!!! یعنی خودمون هر عملیات ریاضی را که خواستیم باید دستی کدشون رو
بزنیم!!!!!!

پاسخ : جاوا زحمت این عملیات ریاضی را کشیده و بصورت آماده در کلاسی به نام **Math** برامون فراهم کرده است.

📍 کلاس **Math** درون پکیج **java.lang** قرار دارد که شامل متدهایی است که عملیات عددی نظیر لگاریتم و توان و....
انجام می دهند.

📍 برای استفاده از متد کلاس **Math** کفایت پکیج آن را درون سورس کد، بالای کلاس خودمون به اصلاح **import**
کنیم:

```
import java.lang.Math;
```

📍 متدهای درون کلاس **Math** همگی استاتیک هستند.

📍 همان طور که میدونید برای استفاده از ویژگی و رفتار های یک کلاس باید از ان شی ساخت ، در یک صورت نیاز به شی
ساختن نیست و مستقیم می توانیم به متغیرها و متدهای کلاس دسترسی پیدا کنیم و آن این هست که متغیرها و رفتار
های کلاس استاتیک باشند.

پس از آنجایی که کلاس **Math** استاتیک هست دیگر برای صدا زدن متدهای آن نیاز به شی ساختن از آن نیست .

مثال: در این مثال چون متدهای کلاس **Math** استاتیک هستند بدون شی ساختن از آن و تنها با بکار بردن مستقیم نام کلاس با حروف بزرگ و نقطه (.) و اسم متد مورد نظر می توان متدهای آن را صدا زد.
در این مثال متد **pow** دو پارامتر ورودی می گیرد، مقدار اول به توان مقدار دوم برده می شود. یعنی ۲ به توان ۴ که جواب ۱۶ می شود.

```
Math.pow(2, 4)
```

مثال: در این مثال مقدار عدد ۲ به توان ۴ در خروجی چاپ می شود:

```
package tutorial;
import java.lang.Math;

public class Power {

    public static void main(String[] args) {


        System.out.println(Math.pow(2, 4));

    }

}
```

خروجی:

```
16.0
```

- دلیل ممیز صفر کنار ۱۶ این هست که متدهای درون کلاس **Math** از نوع اعشاری (**double**) هستند.
- آگه دوست ندارید که عدد ۱۶ بصورت ۱۶٫۰ نمایش داده شود یعنی تمایل دارید که بصورت صحیح نمایش داده شود کافیست عدد اعشار را به عدد صحیح تبدیل کنید!!!! به این کار یعنی تبدیل یک نوع به نوع دیگر را به اصلاح عمل **cast** کردن میگویند!!! اصلا نگران نباشید زیاد سخت نگیرید یک جلسه مفصل در مورد تبدیل نوع ها و عمل **cast** صحبت خواهیم کرد اینجا هم یک مقدمه چینی بود کلا فراموشش کنید 😊
- بخاطر این که سردرگم نشیم مبحث **cast** کردن رو میبیندیم میریم سراغ ادامه موضوع کلاس **Math**  در زیر متدهای پر کاربرد عملیات ریاضی درون کلاس **Math** جمع آوری شده است:

متدهای کلاس Math همراه با توصیف

static double abs(double a)

این متد قدر مطلق یک عدد را محاسبه و برمیگرداند.

همان طور که مشاهده میکنید این متد از نوع اعشاری double میباشد و برخلاف متدهای از نوع void یک مقدار را برمی گرداند نگران نباشید یک جلسه کاملا در مورد متدها بحث خواهیم کرد.

یک پارامتر یا متغیر محلی به عنوان ورودی میگیرد اگر پارامتر مثبت بود مقدار مثبت بر می گرداند اگر منفی بود مقدار مثبت بر می گرداند اگر صفر یا صفر منفی بود صفر مثبت بر می گرداند اگر بی نهایت منفی بود بی نهایت مثبت بر میگرداند.

مثال:

```
package tutorial;
import java.lang.Math;

public class MathDemo {

    public static void main(String[] args) {

        // get some doubles to find their absolute values
        double x = 4876.1874d;
        double y = -0.0d;
        int z=-15;
        // get and print their absolute values
        System.out.println("Math.abs(" + x + ")=" + Math.abs(x));
        System.out.println("Math.abs(" + y + ")=" + Math.abs(y));
        System.out.println("Math.abs(-9999.555d)=" + Math.abs(-9999.555d));
        System.out.println("Math.abs(" + z + ")=" + Math.abs(z));
    }
}
```

خروجی:

```
Math.abs(4876.1874)=4876.1874
Math.abs(-0.0)=0.0
Math.abs(-9999.555d)=9999.555
Math.abs(-15)=15
```

پارامتر ورودی متد **abs** اعشاری هست اما ما بهش عدد صحیح هم دادیم جواب داد!! دلیلش این هست که متد **abs** در کلاس **Math** از نوع های متفاوتی نظیر **int, double, float, long** می باشد!!!!

```
static float abs(float a)
static int abs(int a)
static long abs(long a)
```

این رو بدون که با متد **abs** علاوه بر اعداد اعشاری میشه قدر مطلق اعداد صحیح را هم به دست آورد دیگه نیاز به فلسفه بافی نیست 😊

static double acos(double a)

این متد **arc cosine** (کسینوس معکوس) یک عدد را برمیگرداند.

نکته: برای به دست آوردن عدد پی (π) در جاوا از دستور زیر استفاده میکنیم:

Math.PI

مثال:

```
package tutorial;
import java.lang.*;

public class MathDemo {

    public static void main(String[] args) {

        // get a variable x which is equal to PI/2
        double x = Math.PI / 2;


        // convert x to radians
        x = Math.toRadians(x);

        // get the arc cosine of x
        System.out.println("Math.acos(" + x + ")=" + Math.acos(x));
    }
}
```


خروجی:

```
Math.acos(0.027415567780803774)=1.5433773235341761
```


```
double x = Math.PI / 2;
```

عدد π بر ۲ تقسیم شده است. چون جواب یک عدد اعشاری هست پس X از نوع اعشاری تعریف میکنیم. 

```
x = Math.toRadians(x);
```

در محاسبات مثلثاتی اعداد باید بر حسب رادیان باشند و با استفاده از این متد عددمون را به رادیان تبدیل کردیم. 

```
System.out.println("Math.acos(" + x + ")=" + Math.acos(x));
```

در پایان مقدار کسینوس معکوس عدد X را به دست آوردیم. 

static double asin(double a)

سینوس معکوس یک عدد که بصورت رادیان می باشد بر میگردداند.

محدوده زاویه ای که برمیگرداند بین $\frac{\pi}{2}$ تا $-\frac{\pi}{2}$ می باشد.

مثال:

```
package tutorial;

import java.lang.*;

public class MathDemo {
    public static void main(String[] args) {

        // get a variable x which is equal to PI/2
        double x = Math.PI / 2;

        // convert x to radians
        x = Math.toRadians(x);

        // get the arc sine of x
        System.out.println("Math.asin(" + x + ")=" + Math.asin(x));
    }
}
```

خروجی:

```
Math.asin(0.027415567780803774)=0.02741900326072046
```

static double atan(double a)

تانژانت معکوس یک عدد که بصورت رادیان می باشد بر میگرداند.

محدوده زاویه ای که برمیگرداند بین $\frac{\pi}{2}$ تا $-\frac{\pi}{2}$ می باشد.

مثال:

```
package tutorial;

import java.lang.*;

public class MathDemo {

    public static void main(String[] args) {

        // get a variable x which is equal to PI/2
        double x = Math.PI / 2;

        // convert x to radians
        x = Math.toRadians(x);

        // get the arc tangent of x
        System.out.println("Math.atan(" + x + ") = " + Math.atan(x));
    }
}
```

خروجی:

```
Math.atan(0.027415567780803774)=0.0274087022410345
```

static double cbrt(double a)

ریشه مکعب (توان ۳) عدد را بر می گرداند. مثلا ریشه توان ۳ عدد ۱۲۵ برابر با ۵ یا ریشه مکعب عدد ۲۷ برابر ۳ می باشد.

نکته: $\text{cbrt}(-x) == -\text{cbrt}(x)$

همان طور که می دانید توان ۳ عدد منفی را مثبت نمی کند پس ریشه توان ۳ عدد -۱۲۵ می شود -۵
مثال:

```
package tutorial;

import java.lang.*;

public class MathDemo {

    public static void main(String[] args) {

        // get two double numbers
        double x = 125;
        double y = 10;

        // print the cube roots of three numbers
        System.out.println("Math.cbrt(" + x + ")=" + Math.cbrt(x));
        System.out.println("Math.cbrt(" + y + ")=" + Math.cbrt(y));
        System.out.println("Math.cbrt(-27)=" + Math.cbrt(-27));

    }
}
```

خروجی:

```
Math.cbrt(125.0)=5.0
Math.cbrt(10.0)=2.154434690031884
Math.cbrt(-27)=-3.0
```

• سایر متدهای کلاس Math به شرح زیر می باشد:

static double cos(double a)

این متد کسینوس یک عدد را برمیگرداند.

static double cosh(double x)

کسیونس هایپربولیک عدد را بر می گرداند

static double log(double a)

لگاریتم عدد بر مبنای e را بر می گرداند.

static double log10(double a)

لگاریتم عدد بر مبنای 10 را بر می گرداند.

static double max(double a, double b)

ماکزیمم دو عدد را مشخص و بر میگرداند

static double min(double a, double b)

مینیمم دو عدد را مشخص و بر میگرداند

static double pow(double a, double b)

مقدار عدد اول به توان عدد دوم را بر میگرداند. a^b

static double random()

تولید اعداد تصادفی، بصورت رندم یا تصادفی به ما عدد می دهد.

اعداد تولیدی اعشاری و در بازه ۰/۰ تا کمتر از ۱/۰ می باشد.

- در مورد تولید عدد تصادفی در بازه دلخواه در یک جلسه بهش مفصل می پردازیم.

مثال:

```
package tutorial;

import java.lang.*;

public class MathDemo {
```

```
public static void main(String[] args) {  
  
    // get two random double numbers  
    double x = Math.random();  
    double y = Math.random();  
  
    // print the numbers and print the higher one  
    System.out.println("Random number 1:" + x);  
    System.out.println("Random number 2:" + y);  
    System.out.println("Highest number:" + Math.max(x, y));  
}  
}
```

خروجی:

```
Random number 1:0.5171881476656942  
Random number 2:0.22391213165734392  
Highest number:0.5171881476656942
```

static double sqrt(double a)

جذر عدد را به ما می دهد \sqrt{a}

static double tan(double a)

تانژانت عدد را بر می گرداند $\tan a$

کلاس Math هنوز هم متد داره که ما در اینجا پر کاربردترین آنها را بررسی کردیم.

تا می توانید مثال و تمرین حل کنید خیلی برای یادگیری جواب میده، تئوری کار کردن را رها

کنید مستقیم برید تو گود مطمئن باشید دانش هم بعدش میاد. آرزوی موفقیت 😊

پیروز و موفق باشید

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

www.JAVAPro.ir

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!

بازدید از کانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید