6.13 Math object

Introduction to the Math object

The *Math* object provides properties for mathematical constants and methods to perform mathematical functions.

PARTICIPATION ACTIVITY

6.13.1: Math properties.



Match the Math property to the property's description.

If unable to drag and drop, refresh the page.

Math.PI	Value of π, approximately 3.142	Correct
	π is the ratio of a circle's circumference to the circle's diameter.	
Math.E	Euler's number, approximately 2.718 Euler's number is the base of natural logarithms.	Correct
Math.LN2	Natural logarithm of 2, approximately 0.693 Math.LN10 is the natural log of 10.	Correct
Math.LOG10E	Base 10 logarithm of E, approximately 0.434 Math.LOG2E is the base 2 log of E.	Correct
Math.SQRT2	Square root of 2, approximately 1.414 Math.SQRT1_2 is the square root of 1/2.	Correct

Reset

Feedback?

Math methods

The Math object has a range of trigonometric methods, including sin(), cos(), and tan(), and general calculation methods, including log() and pow(). Some commonly used Math methods are summarized in the table below.

Table 6.13.1: Common Math object methods.

Method	Description	Example
abs(x)	Returns the absolute value of x	Math.abs(-5); // 5
ceil(x)	Returns x rounded up to the nearest integer	Math.ceil(2.1); // 3
cos(x)	Returns the cosine of the radians x	Math.cos(Math.PI) //
floor(x)	Returns x rounded down to the nearest integer	Math.floor(2.9) // 2
log(x)	Returns the natural logarithm of x	Math.log(Math.E) // 1
max(n1, n2, n3,)	Returns the largest number	Math.max(5, 2, 8, 1) // 8
min(n1, n2, n3,)	Returns the smallest number	Math.min(5, 2, 8, 1) // 1
pow(x, y)	Returns x to the power of y	Math.pow(2, 3) // 8
round(x)	Returns x rounded to the nearest integer	Math.round(3.5) // 4
sin(x)	Returns the sine of radians x	Math.sin(Math.PI) // 0
sqrt(x)	Returns the square root of x	Math.sqrt(25) // 5
tan(x)	Returns the tangent of radians x	Math.tan(Math.PI / 4) // 1

Feedback?

PARTICIPATION ACTIVITY

6.13.2: Math methods.

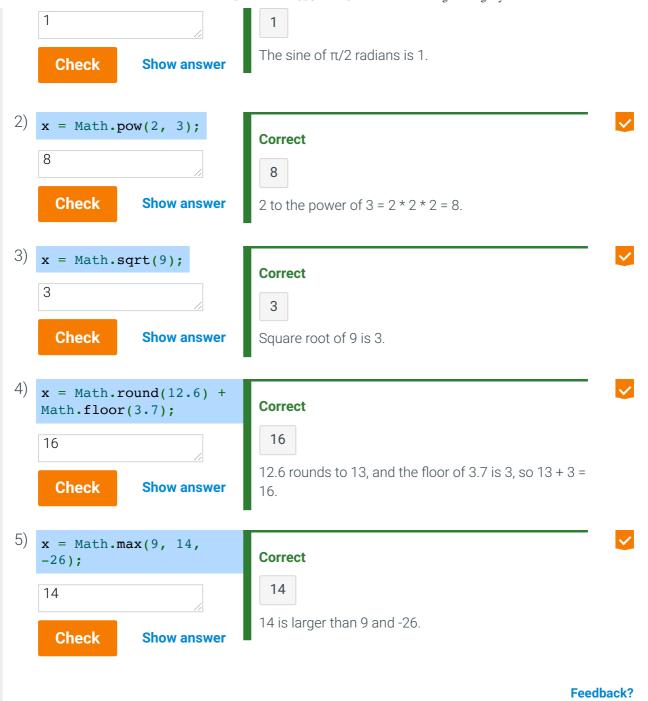


Enter the value assigned to \boldsymbol{x} in each code segment.

1) x = Math.sin(Math.PI /
2);

Correct





Producing random numbers

Many applications, especially games and simulations, need random numbers to simulate random processes. The *Math.random()* method returns a pseudo-random number ≥ 0 and < 1. A *pseudo-random number* is a number generated by an algorithm that approximates randomness, but is not truly random.

Figure 6.13.1: Display 5 random numbers with Math.random().

```
for (let i = 0; i < 5; i++) {
    console.log(Math.random());
}

0.5216294566239728
0.5399290004983317
0.05689844662407162
0.8711941395310085
0.7131957592778093</pre>
```

Feedback?

The figure below shows a <code>getRandomNumber()</code> function that performs the necessary calculations to generate a random integer between two integers.

Figure 6.13.2: Display five random numbers between 1 and 10.

```
// Return a random integer between min and max (inclusive).
function getRandomNumber(min, max) {
   return Math.floor(Math.random() * (max - min + 1)) +
   min;
}

for (let i = 0; i < 5; i++) {
   console.log(getRandomNumber(1, 10));
}</pre>
```

Feedback?

PARTICIPATION ACTIVITY

6.13.3: Random numbers.



 Numbers produced by Math.random() appear to be random.

Correct



pear to be randor

True

False

Math.random() produces pseudo-random numbers that are not truly random but are suitable for many applications.

2) Math.floor(Math.random()

* 10) produces a random

Correct

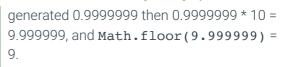


10 is not included. Math.random() returns a number less than 1. If Math.random()

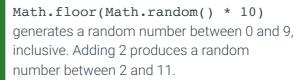
number between 0 and 10, inclusive.

O True

- False
- 3) Math.floor(Math.random()
 * 10) + 2 produces a random
 number between 2 and 11,
 inclusive.
 - True
 - False



Correct





PARTICIPATION ACTIVITY

6.13.4: Practice with random numbers.



The displayCard(rank, suit) function displays a playing card to the console, given the rank (1-13) and suit (0-3).

Write a for loop that calls displayCard() 10 times, each time with a random rank and suit.

```
1 // rank should be a number between 1 and 13, and suit between
 2 function displayCard(rank, suit) {
      switch (rank) {
 3
         case 1: rank = "A"; break;
 4
         case 10: rank = "T"; break;
 5
         case 11: rank = "J"; break;
 6
 7
         case 12: rank = "Q"; break;
         case 13: rank = "K"; break;
 8
         default: if (rank < 1 \mid | rank > 13) {
9
10
            console.log("Bad rank value: " + rank);
11
         }
12
13
14
      switch (suit) {
         case 0: suit = "♥"; break;
15
         case 1: suit = "♦": break:
16
```

Run JavaScript

Reset code

Your console output

▼ View solution

```
• Explain
```

```
--- START FILE: JavaScript ---
function displayCard(rank, suit) {
   switch (rank) {
      case 1: rank = "A"; break;
      case 10: rank = "T"; break;
      case 11: rank = "J"; break;
      case 12: rank = "Q"; break;
      case 13: rank = "K"; break;
      default: if (rank < 1 | | rank > 13) {
         console.log("Bad rank value: " + rank);
      }
   }
   switch (suit) {
      case 0: suit = "♥"; break;
      case 1: suit = "♦"; break;
      case 2: suit = "♣"; break;
      case 3: suit = "\( \black \); break;
      default: console.log("Bad suit value: " + suit);
   }
   console.log("  ");
   console.log(" | " + rank + suit + " | ");
                 |");
   console.log("
   console.log(" | " + rank + suit + " | ");
```

```
console.log(" | ");
}
displayCard(5, 0);
displayCard(13, 1);
displayCard(1, 2);
displayCard(10, 3);
function getRandomNumber(min, max) {
   return Math.floor(Math.random() * (max - min + 1)) + min;
}
for (let i = 0; i < 10; i++) {
   let rank = getRandomNumber(1, 13);
   let suit = getRandomNumber(0, 3);
   displayCard(rank, suit);
}
--- END FILE: JavaScript ---
                                                           Feedback?
```

