

Math.abs()

لما كانت الدالة `abs` هي دالة ساكنة (static method) تابعة للكائن `Math`، فيجب عليك استخدامها دومًا بالشكل `Math.abs(x)`، إذ لا تستطيع استخدامها كجزء من كائن `Math` خاص بك (تذكر أنَّ الكائن `Math` ليس له دالة بانية).

الدالة `Math.abs()` تُعيد القيمة المطلقة للعدد، أي $|x|$ ، وتساوي `x` إذا كانت قيمة `x` أكبر من 0، وتساوي 0 إذا كانت قيمة `x` تساوي 0، وتساوي `-x` إذا كانت قيمة `x` أصغر من الصفر؛ أي في جميع الحالات ستكون القيمة المُعادة هي قيمة موجبة.
ودا لان سالب + سالب = موجب

تمرير كائن فارغ، أو مصفوفة فيها أكثر من رقم، أو سلسلة نصية ليست رقمية، أو متغير فارغ أو قيمته `undefined`، سيؤدي إلى إعادة القيمة `NaN`؛ أما تمرير `null` أو سلسلة نصية فارغة فتُعِيد القيمة 0:

Ex:

```
Math.abs('-1'); // 1
Math.abs(-2); // 2
Math.abs(null); // 0
Math.abs(''); // 0
Math.abs([]); // 0
Math.abs([2]); // 2
Math.abs([1,2]); // NaN
Math.abs({}); // NaN
Math.abs('string'); // NaN
Math.abs(); // NaN
```

تدعم اغلب المتصفحات (internet explorar - chrome - firefox - opera - safari)

Math.sqrt()

أكبر أو تساوي الصفر، فإنَّ هذه x تعيد الجذر التربيعي للعدد، أي لو كانت قيمة الوسيط `Math.sqrt()` الدالة $y^2 = x$ التي تكون أكبر أو تساوي الصفر والتي تُحقِّق المعادلة y الدالة ستُعِيد القيمة

`NaN` سالبةً فسُتُعَاد القيمة x إذا كانت قيمة الوسيط

فيجب عليك استخدامها دومًا بالشكل `Math`، تابعةً للكائن `(static method)` هي دالةٌ ساكنة `sqrt` لَمَّا كانت الدالة ليس له `Math` خاص بك (تذكر أنَّ الكائن `Math` إذ لا تستطيع استخدامها كجزء من كائن `Math.sqrt(x)`، دالةٌ بانية).

Ex :

```
Math.sqrt(9); // 3
Math.sqrt(2); // 1.414213562373095
Math.sqrt(1); // 1
Math.sqrt(0); // 0
Math.sqrt(-1); // NaN
Math.sqrt(-0); // -0
```

تدعم اغلب المتصفحات (internet explorer - chrome - firefox - opera - safari)

Math.pow()

baseexponent. تعيد ناتج رفع الأساس إلى قوة الأس، أي Math.pow () الدالة

بمعني عندي قيمتين قيمة للرقم نفسه و قيمة لانه يضرب ف نفسه كام مرة

Ex :

```
Math.pow(7, 2); // 49
```

```
Math.pow(7, 3); // 343
```

```
Math.pow(2, 10); // 1024
```

الأس هو عددٌ عشري:

```
Math.pow(4, 0.5); // 2 (الجزر التربيعي للعدد 4)
```

```
Math.pow(8, 1/3); // 2 (الجزر التكعيبي للعدد 8)
```

```
Math.pow(2, 0.5); // 1.4142135623730951 (الجزر التربيعي للعدد 2)
```

```
Math.pow(2, 1/3); // 1.2599210498948732 (الجزر التكعيبي للعدد 2)
```

الأس له إشارة:

```
Math.pow(-7, 2); // 49 (يمكن تربيع الأعداد السالبة)
```

```
Math.pow(-7, 3); // -343 (يمكن تكعيب الأعداد السالبة)
```

```
Math.pow(-7, 0.5); // NaN (لا تملك الأعداد السالبة جذرًا تربيعيًا)
```

(لاحظ أنه إذا كانت إشارة الأساس سالبة وكان الأس عددًا عشريًا فستُعاد القيمة NaN دومًا.)

```
Math.pow(-7, 1/3); // NaN
```

تدعم اغلب المتصفحات (internet explorar - chrome - firefox - opera - safari)

Math.max()

```
Math.max([value1[, value2[, ...]]])
```

الدالة `Math.max()` تعيد أكبر قيمة من بين الوسائط المُمرَّرة إليها.

أكبر قيمة من الأعداد المُمرَّرة إلى الدالة، وإذا لم يكن بالإمكان تحويل أحد الوسائط على الأقل إلى قيمة رقمية، فستُعاد القيمة `NaN`.

إذا لم يكن بالإمكان تحويل أحد الوسائط على الأقل إلى قيمة رقمية، فستُعاد القيمة `NaN`.

إذا لم تُمرَّر أية وسائط إلى الدالة، فالنتيجة هي `-Infinity`.

أمثلة عن استخدام الدالة `Math.max()`

```
Math.max(10, 20); // 20
```

```
Math.max(-10, -20); // -10
```

```
Math.max(-10, 20); // 20
```

Math.min()

```
Math.min([value1[, value2[, ...]]])
```

تعيد أصغر قيمة من بين الوسائط المُمرَّرة إليها `Math.min()` الدالة.

`NaN` إذا لم يكن بالإمكان تحويل أحد الوسائط على الأقل إلى قيمة رقمية، فستُعاد القيمة.

`-Infinity` إذا لم تُمرَّر أية وسائط إلى الدالة، فالنتيجة هي.

الشفيرة الآتية تعثر على أصغر قيمة بين `x` و `y` وتُسندُها إلى `z`:

```
; (var x = 10, y = -20; var z = Math.min(x, y
```

تُستخدَم الدالة `min` عادةً للحرص على ألا تتجاوز قيمة ما حدًا معيَّنًا، فمثلاً الشيفرة الآتية:

```
; (var x = f(foo
```

```
} (if (x > boundary
```

```
; x = boundary
```

```
{
```

