**عملگرها**

ما عملگرهای مختلفی را از زمان مدرسه به خاطر داریم. مانند جمع + ، تفریق - ، ضرب \* ، تقسیم و دیگر عملگرها.

در این بخش ما بر روی زوایایی از عملگرها متمرکز می‎شویم که در مدرسه نیاموخته‌ایم.

**اصلاحات : unary (یونِری) – binary (باینری) – operand (اُپِرِند)**

پیش از ادامه بیایید مفهوم این اصطلاحات را بفهمیم.

یک operand – همان چیزی است که عملگرهای برای آنها اعمال می‌شوند. برای نمونه در ضرب 5 \* 2 دو operand داریم. برخی آن را آرگمان (argument) نیز می‌خوانند.

یک عملگر زمانی unary است که فقط یک operand داشته باشد. برای نمونه unary منفی کننده – که علامت یک عدد را برعکس می‌کند.

کد

یک عملگر زمانی binary است که دو operand داشته باشد. همان عملگر منفی کننده در شکل binary هم وجود دارد.

کد

در اصل در اینجا ما در مورد دو عملگر مجزا صحبت کردیم. اولی عملگر unary منفی کننده (که علامت عدد را برعکس می‌کرد) و دیگری عملگر تفریق binary (تفریق دو عدد از هم).

**تلفیق رشته‌ها، عملگر + binary**

معمولا از عملگر + برای جمع اعداد استفاده می‌شود، اما زمانیکه این عملگر به رشته‌‌‌ها اعمال شود، آنها را ادغام می‌کند.

کد

در نظر داشته باشید که اگر یکی از operand ها string باشد، دیگری نیز به string تبدیل می‌شود.

برای نمونه :

کد

همانطور که می‌بینید مهم نیست که operand ابتدا آمده باشد یا نه، قاعده ساده است: اگر هر کدام از operand ها string بود، دیگری نیز به string تبدیل می‌شود.

با این حال در نظر داشته باشید که عملیات از چپ به راست انجام می‌شود. اگر دو عدد در پیش از یک string آمده باشند، ابتدا آن دو عدد جمع می‌شوند و سپس به string تبدیل می‌شوند.

کد