

Clojure

Sözdizimi (Syntax)

Yazılı değerler (*Literals*)

Aşağıda Clojure'da bulunan temel tiplerin (*primitives*) yazılı temsil (*literal representation*) örneklerini bulabilirsiniz.

`;` karakteri satır sonuna kadar yorum oluşturur. Yaygın bir uygulama olarak yorum bölüm başlıklarını belirlemek için çok sayıda noktalı virgül kullanılabilir, ancak bu teknik bir zorunluluk değildir.

Sayısal tipler

```
42      ; integer
-1.5    ; floating point
22/7    ; ratio
```

Tamsayılar uygun değer aralığında ise 64-bit sabit hassasiyetli (*fixed precision*) tamsayılar olarak, aksi halde isteğe göre hassasiyetli (*arbitrary precision*) tamsayılar olarak okunur. İsteğe göre hassasiyeti tercih etmek için sayının sonuna **N** eklenebilir. Clojure ayrıca sekizli (0 önekli), onaltılı (0x önekli) ve özel tabanlı (taban ve r önekli, ör. ikili (*binary*) taban için 2r) tamsayılar için Java sözdizimini destekler. Ayrıca pay ve paydanın bir araya getirilmesi ile oran (*ratio*) tipi oluşturularak kullanıma sunulmuştur.

Küsürlü (*kayan noktalı/floating point*) değerler çift-hassasiyetli, yahut **M** son eki ile isteğe göre hassasiyetli 64-bit *float*lar olarak okunur. Üslü yazım (*exponential notation*) ayrıca desteklenir. `##Inf`, `##-Inf` ve `##NaN` özel sembolleri sırası ile artı sonsuz, eksi sonsuz ve "sayı değil" değerlerini temsil eder.

Karakter tipleri

```
"hello"  ; string
\e       ; character
#"[0-9]+" ; regular expression
```

Metin değerleri (*string*) çift tırnak içinde yazılır ve birden fazla satır alabilir. Tek karakterler ters eğik çizgi (*backslash*) ile temsil edilir. Özel isimli birkaç karakter: `\newline` `\space` `\tab` vs. Unicode karakterler `\uNNNN` veya `\oNNN` gibi sekizli olarak temsil edilebilir.

Düzenli ifadeler (*literal regular expressions*) `#` ile başlayan metinler olarak yazılır. Bunlar `java.util.regex.Pattern` nesnelere çevrilirler.