

به نام خدا

آموزش نرم افزار CLIPS

به زبان ساده – قسمت اول

WWW.TEHRANIT.NET

برای شروع روی فایل CLIPSWin.exe دابل کلیک کنید. در حال حاضر شما یک پنجره خالی (Dialog Window) که دستور >CLIPS در آن قرار دارد را می بینید. اینجا همان جایی که شما باید دستورات و برنامه خود را بنویسید. به عنوان مثال برای خروج سریع از CLIPS می توانید در اینجا (exit) را تایپ کنید و اینتر را بزنید.



نکته: توجه داشته باشید تمام دستورات CLIPS داخل پرانتز قرار میگیرند.

Facts

پایه ترین عملیات CLIPS روی یک لیست از حقایق (facts) و مجموعه ای از قوانین (rules) کار می کند. یک fact یک قطعه اطلاعات مانند (color green) یا (parent_of Jack Rose) است.

- دستور **assert** : حقایق بوسیله دستور assert ایجاد می شوند و سپس fact ایجاد شده بدرون پایگاه داده حقایق می رود.

مثال: ایجاد یک fact و جواب بازگشتی از CLIPS

```
CLIPS> (assert (color green))
```

```
==> f-1 (color green)
```

```
<Fact-1>
```

قسمت f-1 ==> از جواب بازگشتی از CLIPS نشون دهنده اینه که شماره fact ثبت شده ۱ است (هر fact یک شماره ایندکس دارد).

نکته: برای مشاهده fact ها از منوی windows گزینه Facts Windows را فعال کنید.

همچنین می توان چند fact را باهم ایجاد نمود.

```
CLIPS> (assert (color red) (color blue) (color yellow))
```

```
==> f-2 (color red)
```

```
==> f-3 (color blue)
```

```
==> f-4 (color yellow)
```

```
<Fact-4>
```

- دستور **(facts)** : این دستور تمام حقایق جاری را لیست می کند.

CLIPS> **(facts)**

f-0 (initial-fact)

f-1 (color green)

f-2 (color red)

f-3 (color blue)

f-4 (color yellow)

For a total of 5 facts.

همچنین می توانید از فرمت زیر برای محدود کردن لیست fact ها استفاده نمود:

([بزرگترین ایندکس | ایندکس پایان] ایندکس شروع **(facts)**)

مثال:

CLIPS> **(facts 1 3)**

f-1 (color green)

f-2 (color red)

f-3 (color blue)

For a total of 3 facts.

یعنی حقایق از شماره ۱ تا ۳ رو لیست کن (اگه ۳ رو نمی نوشتیم از ۱ تا آخر رو لیست میکرد).

- دستور **retract** : حقایق رو میشه توسط این دستور حذف کرد. جلوی دستور retract شماره حقایقی که میخوایم حذف کنیم را می نویسیم.

CLIPS> **(retract 1 2)**

<== f-1 (color green)

<== f-2 (color red)

شماره حقایق حذف شده دیگه مورد استفاده قرار نمی گیره. یعنی اگر شما بخواید یک fact جدید ایجاد کنید شمارش ۱ نخواهد بود بلکه ۵ میشه.

- دستور **(clear)** : تمام قوانین و حقایق را از حافظه حذف می کند. معادل بستن و شروع مجدد CLIPS .

برای توسعه یک برنامه که قادر به انجام برخی عملکردهای مفید باشد باید از قوانین استفاده کنید. بطور کلی، یه قاعده (Rule) با فرمت زیر بیان می شود.

'IF something is true THEN do some action'

یعنی اگر شرط درست بود آنگاه یک عملی رو انجام بده

• دستور **defrule**: تعریف قاعده با استفاده از این دستور انجام می شود.

["توضیحات" نام قاعده (defrule)

pattern یا شروط

=> آنگاه

(عملیات در صورت درست بودن شروط

مثال: با یک مثال نحوه تعریف قاعده را توضیح می دهیم.

```
CLIPS> (defrule dog
  (animal-is dog)
=>
  (assert (sound-is woof)))
```

یک قاعده از سه قسمت تشکیل شده. قسمت اول، **(defrule dog)**، یک نام یکتا (در اینجا dog) برای قاعده مشخص می کند. قسمت دوم، **(animal-is dog)**، یک الگو از قاعده است (همان قسمت IF) و قسمت سوم، **(assert (sound-is woof))**، که نشان دهنده عمل است (همان قسمت THEN).

به زبان ساده rule بالا به این معنی است که 'اگر حقیقت **(animal-is dog)** در پایگاه حقایق وجود داشته باشد، آنگاه حقیقت دیگر، یعنی **(animal-is woof)**، را به درون پایگاه حقایق اضافه کن'. امتحان کنید! سیستم رو پاک کنید،

```
CLIPS> (clear)
```

سپس یک حقیقت تعریف کنید

```
CLIPS> (assert (animal-is dog))
```

```
<Fact-1>
```

و بعد آن قاعده را بدقت تایپ کنید و اینتر را بزنید.

```
CLIPS> (defrule dog
  (animal-is dog)
=>
  (assert (sound-is woof) )
CLIPS>
```

در حال حاضر شما یک قاعده تعریف کرده‌اید و هنوز آن را اجرا نکرده‌اید. قاعده تعریف شده را می‌تونید در پنجره agenda مشاهده کنید، همچنین می‌تونید دستور **(agenda)** را تایپ کنید.
نکته: برای مشاهده پنجره agenda از منوی windows گزینه Facts Windows را فعال کنید.

- دستور **(rules)** : این دستور نیز تمام قوانین جاری را لیست می‌کند.

```
CLIPS> (rules)
dog
For a total of 1 defrule.
```

- دستور **(run)** : توسط این دستور می‌تونید قوانین رو اجرا کنید.

```
CLIPS> (run)
==> f-2 (sound-is woof)
CLIPS> (facts)
```

این دستور قوانین شما را اجرا می‌کند. اگه به پایگاه حقایق نگاه کنید می‌بیند که یک حقیقت یعنی (sound-is woof) اضافه شده و این یعنی شرط قاعده شما برقرار بوده که قسمت عملیات اجرا شده است.

نکته: شما می‌تونید در تعریف قوانین چند شرط بگذارید و یا چندین عمل رو بخواهید که در صورت برقرار بودن شرط انجام بشه.

مثال :

```
CLIPS> (defrule is-it-a-duck
  (animal-has webbed-feet)
  (animal-has feathers)
=>
  (assert (animal-is duck)))
(printout t "it's a duck" crlf))
```

در مثال بالا قاعده ای با نام **is-it-a-duck** (این یک اردک است) تعریف شده است و در قسمت IF دو شرط **(animal-has webbed-feet)** و **(animal-has feathers)** (یعنی اگر حیوان پا دارد و اگر حیوان پر دارد) مورد بررسی قرار می‌گیرد که آیا در پایگاه حقایق وجود دارند یا نه؟ در صورت وجود هر دو سپس حقیقت جدیدی با دستور **(assert (animal-is duck))** (حیوان اردک است) اضافه شده و درخط آخر یعنی **(printout t "it's a duck" crlf)** پیام **it's a duck** توسط دستور **printout t** چاپ می‌شود و **crlf** بنظور رفتن به خط بعدی بعد از چاپ پیام استفاده شده است.

