

TAUPROLOG

JAFH

Que es?

Es un interprete de prolog implementado en JS por lo que se ejecuta en el cliente, trabaja con el estándar ISO Prolog.

Obtener Tau Prolog

Tau Prolog es libre y lo puede obtener desde:

<http://tau-prolog.org/>

Esto descarga el archivo: tau-prolog.js

Que después puede cargar en su archivo html así:

```
<script src="tau-prolog.js"></script>
```

Ejemplo

Suponga que tiene los siguientes hechos en una base de conocimiento (BC):

Ana programa en c expresado en prolog como: `programa(ana,c)`

Juan programa en java expresado en prolog como: `programa(juan,java)`

Ana programa en html expresado en prolog como: `programa(ana,html)`

Debe tener la base de conocimiento escrita en su código html así:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="tau-prolog.js"></script>
</head>
<body>
  <div id="respuesta"></div>
</body>
<script id="habilidades.pl" type="text/prolog">
  certificado(ana,c).
  certificado(juan,java).
  certificado(ana,html).
</script>
<script>
</script>
  //codigo de sus consultas
</html>
```

Todos los métodos de tau-prolog estarán cargados en el objeto pl. Para empezar a trabajar con tauprolog(TP) debe crear una sesión:

```
<script>  
  //codigo de sus consultas  
  var session = pl.create();  
</script>
```

Cargar la BC

A continuación debe cargar la BC:

```
session.consult("habilidades.pl");
```

Donde `habilidades.pl` es la base de conocimiento con que trabajará.

```
<script>  
  //codigo de sus consultas  
  var session = pl.create();  
  session.consult("habilidades.pl");  
</script>
```

Cargar la pregunta

El siguiente paso es cargar la pregunta que se realizará en la BC:

```
session.query("programa(juan,java)."); /*preguntamos si juan programa en java*/
```


Cargar la BC

A continuación debe cargar la BC:

```
session.consult("habilidades.pl");
```

Donde `habilidades.pl` es la base de conocimiento con que trabajará.

```
<script>
//codigo de sus consultas
var session = pl.create();
session.consult("habilidades.pl");
session.query("programa(juan,java).");
session.answer(function( answer ) {
    let r=pl.format_answer( answer );
    console.log( "respuesta: "+ r);
    document.getElementById("respuesta").innerHTML="la respuesta es "+r;
});
</script>
```

Query solo carga la pregunta no la ejecuta. A continuación para ejecutar la pregunta utilizamos la función ***answer***. Esta función es asíncrona por lo que no devuelve un resultado como tal, si no que ejecuta una función callback que se le envía como parámetro.

Funciones asíncronas en JS

Una función asíncrona es aquella que se ejecuta en paralelo con nuestro código, es decir se lanza la ejecución de la función y sin esperar que termine se continua con nuestro código, la función esta trabajando mientras nuestro código se sigue ejecutando, esto crea el problema de ¿como saber cuando la función terminó y su respuesta esta lista?, pues para esto, a este tipo de funciones se les da como parámetro una función del tipo callback, que se ejecutará solo cuando la función asíncrona termine y este lista su repuesta. De manera que este código nos esta diciendo:

```
session.answer(function(respuesta){  
    //Hacer algo con la respuesta  
});
```

Session.answer, ejecútate y cuando tengas listo el resultado ejecuta la función callback que te estoy enviando como parámetro:

```
function(respuesta){  
    //Hacer algo con la respuesta  
}
```

Enviándole como parámetro tu respuesta.

La función callback

La función callback que se esta enviando como parámetro es **esta**:

```
session.answer(function( answer ) {  
    let r=pl.format_answer( answer );  
    console.log( "respuesta: "+ r);  
    document.getElementById("respuesta").innerHTML="la respuesta es "+r;  
});
```

Esta función recibe el parámetro **answare** enviado por la función **session.answer**, el parámetro answare es un objeto del tipo ***substitution*** definido por tau-prolog que contiene la respuesta de la consulta. La función callback, esta usando el método **format_answer** del objeto pl, este método recibe como parámetro un objeto del tipo ***substitution*** y lo convierte a texto.

Código completo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="tau-prolog.js"></script>
</head>
<body>
  <div id="respuesta"></div>
</body>
<script id="habilidades.pl" type="text/prolog">
  programa(ana,c).
  programa(juan,java).
  programa(ana,html).
</script>
<script>
  //codigo de sus consultas
  var session = pl.create();
  session.consult("habilidades.pl");
  session.query("programa(juan,java).");
  session.answer(function( answer ) {
    let r=pl.format_answer( answer );
    console.log( "respuesta: "+ r);
    document.getElementById("respuesta").innerHTML="la respuesta es "+r;
  });
</script>
</html>
```

Referencias:

<http://tau-prolog.org/manual/es/un-tutorial-sencillo>