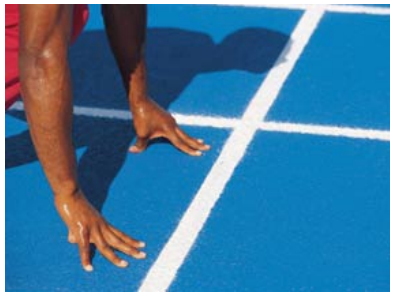
maXbox



maXbox Starter 72

Multilanguage Coding

1.1 Common Gateway Coding (CGC)

Software is changing the world. CGC is a Common Gateway Coding as the name says, it is a "common" language approach for almost everything.
I want to show how a multi-language approach to infrastructure as code using general purpose programming languages lets cloud engineers and code producers unlocking the same software engineering techniques commonly used for applications.
Quite simply, CGC stands for ***Common Gateway Coding***. So, CGC is the **API** for programming applications and services.
It is so trivial and naïve from the name, but let us start with 10 short examples in the name of different scripts for a Weather Station:

HTML, XML, UML, Pascal, Python, Regex, JavaScript, SQL, PHP and Delphi.

First we start with HTML, JavaScript and CGI:

```
program CGI1;

{$mode objfpc}{$H+}  {$APPTYPE CONSOLE}
uses
 {$IFDEF UNIX}{$IFDEF UseCThreads}
 cthreads,
 {$ENDIF}{$ENDIF}
 Classes, SysUtils, CustApp   { you can add units after this };
  begin
   writeln('content-type: text/html');
   writeln('');
   writeln('<html>');
   writeln('<head>');
   writeln('<title>HTML CGI Demo</title>');
   writeln('<meta http-equiv="refresh" content="5">');
```

```
  writeln('</head>');
  writeln('<body>');
  writeln('*********************************************************************************');
  writeln(' <h1>Welcome to OpenWeatherMap SSL & maXbox4 scripts</h1>'       );
  writeln('*********************************************************************************');
  writeln('<br>');
  writeln('Hello, CGC maXbox world!');
  writeln('<br>');
  writeln(dateTimetoStr(now));
  writeln('</body>');
  writeln('</html>')
end.
```

The interesting one in above script is the calling of a compiled function `writeln(dateTimetoStr(now));` which is an internal function of your language, so we don't need Java Script but we can!

So we go on with Java Script. Remember also: The server and the client (the browser) usually run on different computers. They may run under different operating systems, even with different languages.

```
function _MakeExecution(const aCode: AnsiString): AnsiString;
 var aJavaScript: OleVariant;
  begin
    aJavaScript:= CreateOleObject('ScriptControl');
    aJavaScript.Language:= 'JavaScript';
    result:= AnsiString(aJavaScript.Eval(String(aCode)));
  end;

begin
   writeln('runsxipt: '+runJS('Math.pow(2, 128)'))
   //writeln('runsxipt: '+runJS('var strint = require("./strint")'))
   writeln('runsxipt: '+runJS('var d = new Date(); n = d.getDate();'))
   writeln('runsxipt: '+runJS('var d = new Date(); n = d.getTime();'))

      writeln(runjs('Math.pow(4.6,3)'))
      writeln(runjs('Math.sin(-0.9)'));
      maxcalcf('sin(-0.9)');
      writeln(floattostr(sin(-0.9)))
      sr:= 'var totn_string = "Tech On The Net";';

      writeln(runjs('var totn_string = "Tech On The Net"; '+
                                  'totn_string.split(" "); '+
                                  'totn_string.split('' '', 2)'))
      writeln(runjs(sr+ 'totn_string.split(" "); '));
      writeln(runjs(sr+ 'totn_string.split(" "); '+
                     'for (var counter = 1; counter < 5; counter++) {'+
                     '"totn_string"}'));
```

Most data is available in JSON, XML, or HTML format, so XML and HTML is also a format language. The browser really asks for a "resource" and does not know, or care, where the server gets the data from. First thing we have to do is to compile the script above .

Then we put the exe in the cgi_bin directory or configure in apache:

```
<form action="C:\WWW\cgi-bin\CGI1.exe " method="get">
```

As you see I did a refresh in the CGI script so we can see every 5 seconds a GET -request received from:

```
writeln('<meta http-equiv="refresh" content="5">');
```

Next we jump to XML as a markup language:

```
<city id="2643743" name="London">
  <coord lon="-0.13" lat="51.51"/>
  <country>GB</country>
  <timezone>3600</timezone>
  <sun rise="2019-09-23T05:47:32" set="2019-09-23T17:58:36"/>
</city>
```

Once we have the XML representation of a tree view and his items, we can use it to populate the tree view. When the application starts, the `XML2Tree` procedure is called to construct the tree. The tree parameter is a reference to a `TTreeView` component we are populating; the XMLDoc parameter points to a `TXMLDocument` component. In this case we are using the `TXMLDocument` component dropped on a form.

```
procedure TestWithXPath;
var Doc, Nodes, ChildNodes, Node : OleVariant;
    i: integer;
    s: string;
begin
  // Create a COM object If you have MSXML 4:
  //Doc := Sys.OleObject('Msxml2.DOMDocument.4.0'); // If MSXML 6:
  Doc:= CreateOleObject('Msxml2.DOMDocument.6.0');
  Doc.async:= false;
  // Load data from a file
  // We use the file created earlier
  Doc.load(XMLDOCF);

// Report error, if for instance, markup or file structure is invalid
  if Doc.parseError.errorCode <> 0 then begin
    s := 'Reason:' + #9 + Doc.parseError.reason + #13#10 +
         'Line:' + #9 + VarToStr(Doc.parseError.line) + #13#10 +
         'Pos:' + #9 + VarToStr(Doc.parseError.linePos) + #13#10 +
         'Source:' + #9 + Doc.parseError.srcText;
```

```
    // Post an error to the log and exit
    writeln('Cannot parse the document. '+ s);
    Exit;
  end;
```

In modern times, a configuration file has to be an XML standard so you want to parse that file to get the elements from corresponding nodes.

We just start our compiled CGI with the following URL:

http://192.168.1.66/cgi-bin/CGI1.exe

The code object is an Indy Extension called `TidCGIRunner`:

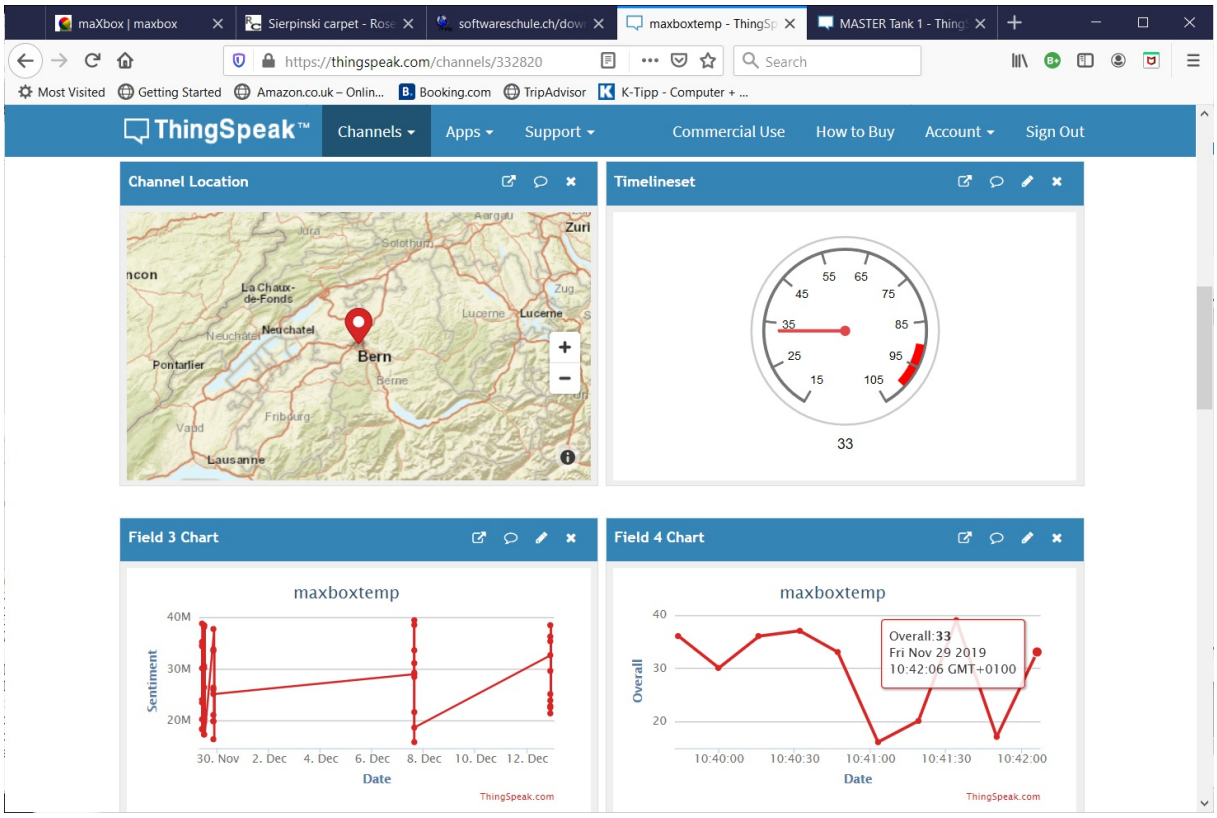
```
//object CGIRunner: TidCGIRunner
  CGIRunner:= TidCGIRunner.create(self)
    with CGIRunner do begin
      Server := HTTPServer
      TimeOutMsg := '<html><body><h1><center>Process is
                                terminated.</body></html>'#13#10
      ErrorMsg := '<html><body><h1><center>Internal Server
                                Error</body></html>'#13#10
      ServerAdmin := 'admin@server'
      PHPSupport:= true;
      //RegisterProperty('PHPIniPath', 'String', iptrw);
      //Left := 260
      //Top := 125
  end //}
```

This means that in order to save "state-information", we must do something special. In fact, there are three common ways to safe state information: this is a Regex to get state-information from the OpenWeather API:

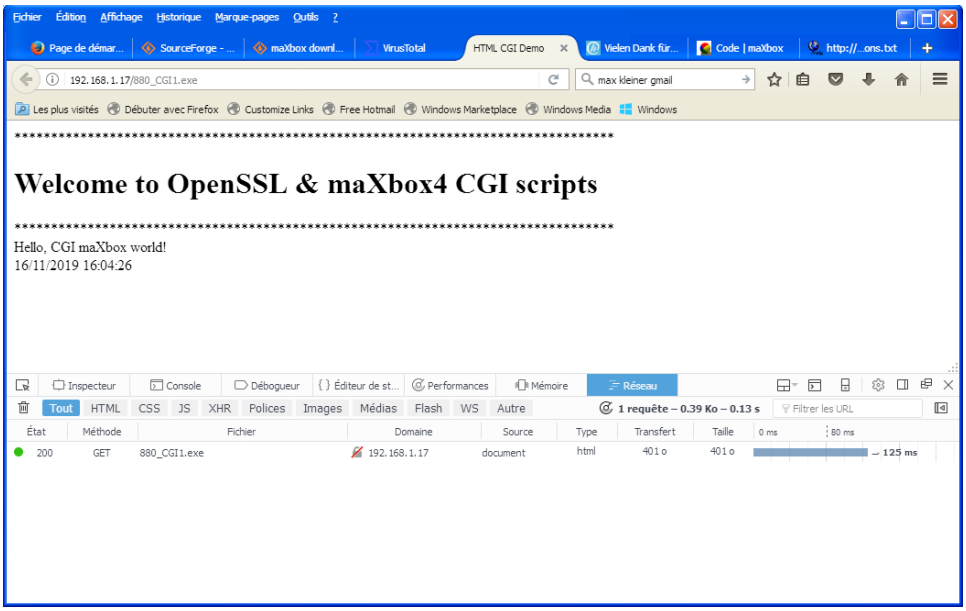
```
Const
 WeatherREX2 =
    '"id":([\w]+).*"main":"([\w\s]+)".*"description":"([\w\s]  //1-4
                                +)".*"temp":([\d\.\-]+).*'+
    '"temp_min":([\d\.\-]+).*"temp_max":([\d\.\-]+).*'+       //5+6
    '"pressure":([0-9]+).*'+                                  //7
    '"humidity":([0-9]+).*'+                                  //8
    '"country":"([\w\s]+).*"name":"([\w\s`]+)"';              //9+10
```

If you do not see some of the parameters in your API response it means that these weather phenomena are just not happened for the time of measurement for the city or location chosen. Only really measured or calculated data is displayed in API response!
Next we put all this together with Things Speak. ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak

from your devices, create instant visualization of live data, and send alerts or aggregated data[1]. One reason you might note is that you send the data with a script from the shell or a console like maXbox.



One reason you might note is that a CGI application is executed as a separate process every time a web page requests it. This is unlike some forms of Apache modules or ISAPI IIS DLLs where the CGI application is loaded once and stays in memory.



[1] https://thingspeak.com/

Another reason you may avoid using CGI is that the burden for security is higher with CGI than it is with a server-side scripting language.

Here comes SQL with WMI to get the logs of the weather station.

```
isloc:= WMIStart;
iserv:= WMIConnect(isloc, 'localhost','','');
aQuery:= 'Select * from Win32_NTLogEvent Where Logfile ="System"
                    And EventType= "1"'+
                     ' AND User = "NT AUTHORITY\\NETWORK SERVICE"';
 it:= 0;
 iset:= WMIExecQuery(iserv, aQuery);
 if WMIRowFindFirst(iset, ENum, vObj) then repeat
   inc(it)
   PrintF('-systime %d: %s - Name %s User %s'+#13#10+'%s'+#13#10,
             [it,vObj.TimeGenerated, vObj.SourceName,
                  vObj.User,vObj.message])
 until not WMIRowFindNext(ENum, vObj);
 writeln('System Network Errors : '+itoa(it))
 vObj:= unassigned;
 vObj:= Null;
```

The SQL SELECT statement returns a result set of records from one or more tables. If you want to fetch all the fields available in the field, then you use the * syntax.
There are limitations in scripting languages that are not present in your own application, but Pascal, Python or PHP are rich of function features.

PHP Now: Because of that, extra caution needs to be applied, especially in how you handle input coming from (you hope) your users. Check out the section on security to learn about what methods are used to make your CGI app tight.

https://www.codeproject.com/articles/9433/understanding-cgi-with-c

```
//object HTTPServer: TIdHTTPServer
    HTTPServer:= TIdHTTPServer.create(self)
    with httpserver do begin
     HTTPServer.bindings.Clear;
     HTTPServer.OnStatus:= @TfmHTTPServerMainHTTPServerStatus;
     defaultport:= PORT_Socket; //8090 Or 4443;
     //Bindings.Add;
       bindings.add.port:= defaultport;
       bindings.add.ip:= IP_Socket;
```

So, we create a own CGI runner and It works well with ThingSpeak, if I set the working directory to child process created for my CGI runner executing a compiled server side scripting **CGI1.exe**.

By default, PHP is built as both a CLI and CGI program, which can be used for CGI processing. If you are running a web server that PHP has module support for, you should generally go for that solution for performance reasons. However, the CGI version enables users to run different PHP-enabled pages under different user-ids. You can find the script-code at:

http://www.softwareschule.ch/examples/cgi.txt

The whole project with source and compiled exe at:

https://sourceforge.net/projects/maxbox/files/Examples/13_General/880_cgiMain_server3.pas/download

In the end of our language guide tour we show a Python SSL Server to enhance our Script Server:

For a test suite we need to create a local SSL-enabled HTTPS server in my Python project. Also, Python has shipped its own built-in SSL module for quite a while.

```
from http.server import HTTPServer, SimpleHTTPRequestHandler
import ssl

httpd = HTTPServer(('localhost', 4443), SimpleHTTPRequestHandler)
httpd.socket = ssl.wrap_socket(httpd.socket,
                               certfile='/tmp/cert_key.pem', server_side=True)
httpd.serve_forever()
```

You got it to work with a remarkably simple piece of code using the builtin ssl module:

```
from http.server import HTTPServer, SimpleHTTPRequestHandler
import ssl
```

(We use port 4443 so that I can run the tests as normal user; the usual port 443 requires root privileges).

The integrated key file can be genrated with cat or type in the shell:

>>> type file1 file2 > file3

is equivalent of:

$ cat file1 file2 > file3

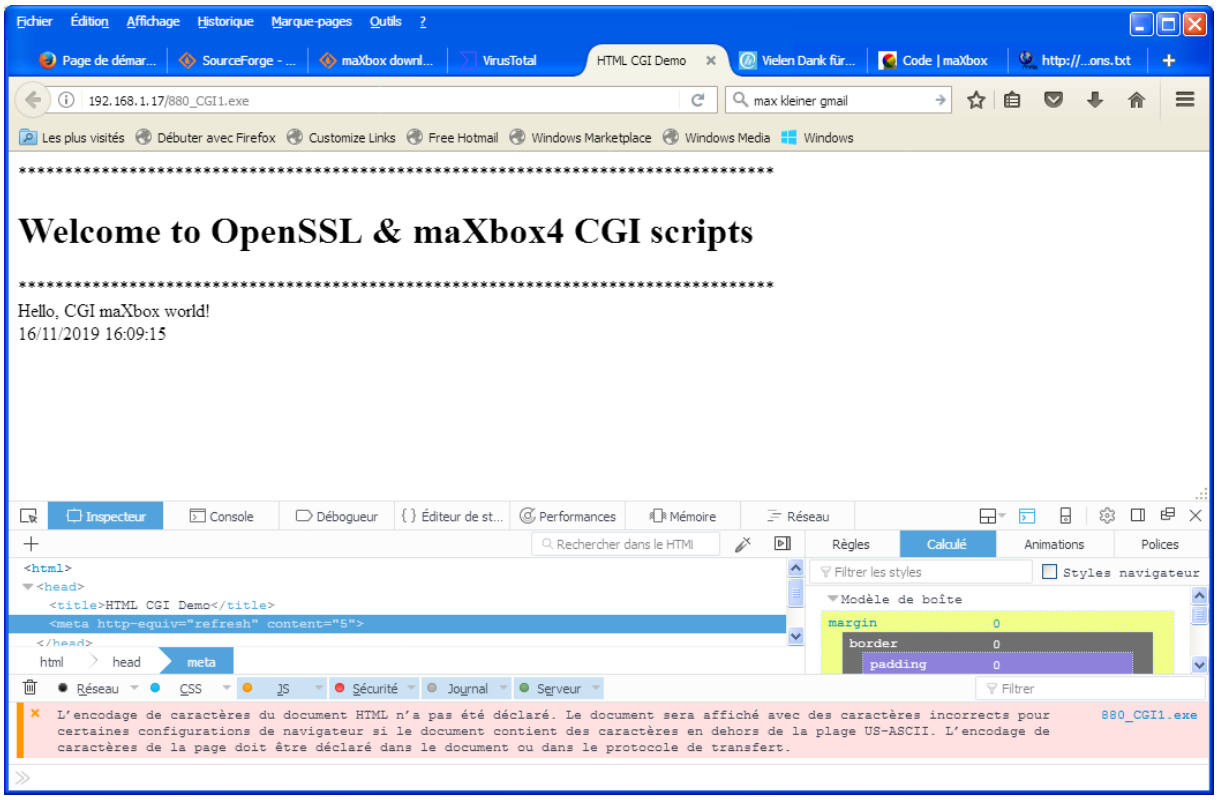
28/11/2019 17:44 <DIR> ..
27/11/2019 14:04 1,716 CA_crt.pem
27/11/2019 15:24 4,711 CERT_host_softwareschule_crt.pem
27/11/2019 15:22 1,751 PVK_host_softwareschule.pem
3 File(s) 8,178 bytes

so we have to stick together the certificate and private key to ONE file:

>>> type CERT_host_softwareschule_crt.pem PVK_host_softwareschule.pem > cert_key.pem

So you have seen CGI and CGC in all sorts of places on the Web, although you may not have known it at the time. For example:

- Any guest book allows you to enter a message in an HTML form and then, the next time the guest book is viewed, the page will contain your new entry.
- The WHOIS form at Network Solutions allows you to enter a domain name on a form, and the page returned is different depending on the domain name entered.
- Any search engine lets you enter keywords on an HTML form, and then it dynamically creates a page based on the keywords you enter.



https://www.bytesin.com/software/maXbox/

1.2 Web Sockets with Pascal and Delphi

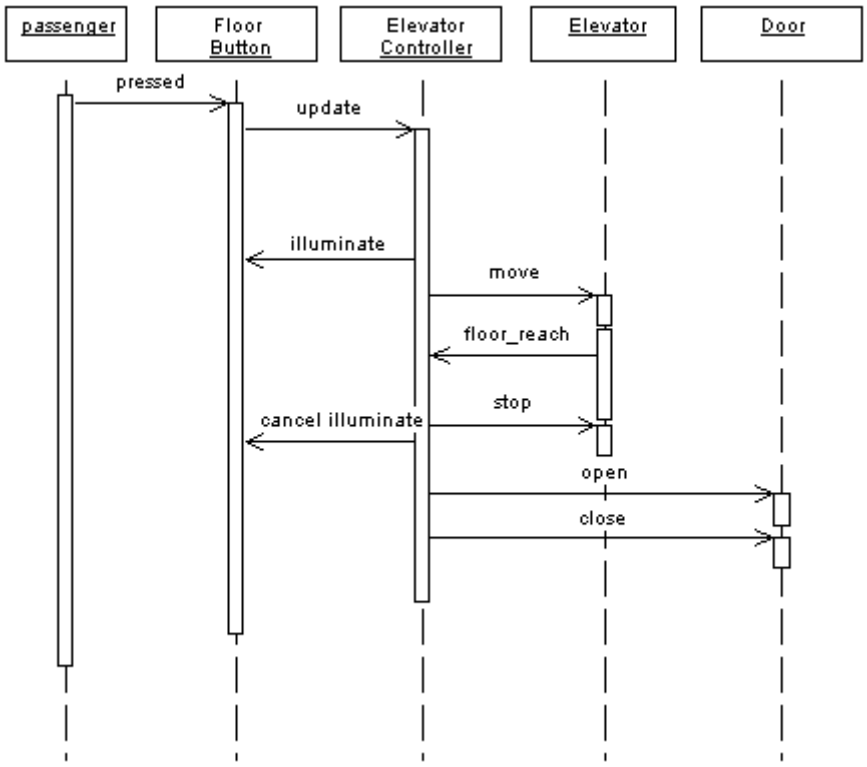
- According to the draft specification, the Web Socket protocol enables two-way communication between a user agent running untrusted code running in a controlled environment to a remote host that has opted-in to communications from that code. The security model used for this is the Origin-based security model commonly used by Web browsers. The protocol consists of an initial handshake followed by basic message framing, layered over TCP. The goal of this technology is to provide a mechanism for browser-based applications that need two-way communication

with servers that does not rely on opening multiple HTTP connections (e.g. using XMLHttpRequest or iframes and long polling).

- Translated into human language, using Web Sockets, the (web)server can now initiate communication and send data to the client (browser) without being asked. This happens after a trusty channel of communication is established over a TCP connection.

```
// The TWebSocketConnection represents the communication
 // chanel with a single connected client. When the connection
 // is innitiated, the handshake procedure is executed. If the
 // handshake succedes the channel can be used for sending and
 // receiving. If not, the connection must be closed.

 TWebSocketConnection = class
 private
   FPeerThread: TIdPeerThread;
   FHandshakeRequest: TWebSocketRequest;
   FHandshakeResponseSent: Boolean;
   FOnMessageReceived: TWebSocketMessageEvent;
   function GetHandshakeCompleted: Boolean;
   function GetServerConnection: TIdTCPServerConnection;
   function GetPeerIP: string;
 protected
   const
     FRAME_START = #$00;
     FRAME_SIZE_START = #$80;
     FRAME_END = #$FF;

   procedure Handshake;
   procedure SendFrame(const AData: string);

   property PeerThread: TIdPeerThread read FPeerThread write FPeerThread;
   property ServerConnection: TIdTCPServerConnection read
                                                 GetServerConnection;
   property HandshakeRequest: TWebSocketRequest read FhandshakeRequest
                                        write FHandshakeRequest;
   property HandshakeCompleted: Boolean read GetHandshakeCompleted;
   property HandshakeResponseSent: Boolean read FhandshakeResponseSent
                                    write FHandshakeResponseSent;
 public
   constructor Create(APeerThread: TIdPeerThread);

   procedure Receive;
   procedure Send(const AMessage: string);

   property OnMessageReceived: TWebSocketMessageEvent read
                               FOnMessageReceived write FOnMessageReceived;
   property PeerIP: string read GetPeerIP;
 end;
```

In the end we got the language UML. UML is said to address modelling of manual, as well as systems, and to be capable of describing current systems and specifying new ones. UML is intended analysis and design language, not a full software development methodology. It specifies notations & diagrams, but makes no suggestions how these should be used in dev process.
For example in the following sequence-diagram a passenger is the browser with a floor button. The elevator is the server and after handshaking a communication the server pushes commands like illuminate to the floor.



A WebSocket server is an application listening on any port of a TCP server that follows a specific protocol, simple as that. The task of creating a custom server tends to scare people; however, it can be easy to implement a simple WebSocket server on your platform of choice.

```
// The TWebSocketServer is a TCP server "decorated" with
  // some websocket specific functionalities.
  TWebSocketServer = class
  private
    FDefaultPort: Integer;
    FTCPServer: TIdTCPServer;
    FThreadManager: TIdThreadMgr;
    FConnections: TObjectList;
    FOnConnect: TWebSocketConnectEvent;
    FOnMessageReceived: TWebSocketMessageEvent;
    FOnDisconnect: TWebSocketDisconnectEvent;
    function GetTCPServer: TIdTCPServer;
    function GetThreadManager: TIdThreadMgr;
```

```
function GetConnections: TObjectList;
function GetActive: Boolean;
procedure SetActive(const Value: Boolean);
```

Code and script can be found at:
http://www.softwareschule.ch/examples/751_Elevator_Simulator4.pas

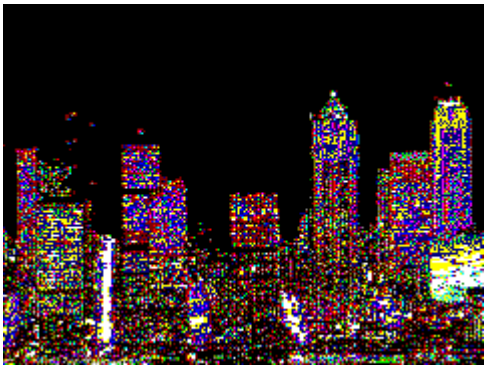
http://www.softwareschule.ch/examples/766_wmi_management.txt

Conclusion:

The handshake is the "Web, Edit and Script" in WebSockets. It's the bridge from HTTP to WebSockets. In the handshake, details of the connection are negotiated, and either party can back out before completion if the terms are un-favorable. The server must be careful to understand everything the client asks for, otherwise security issues will be introduced.

if you are working with an existing PHP or Python installation which did not build either the commandline or CGI servers, you can use the lynx non-graphical web browser to get the web server to execute php scripts from the command line (or cron jobs, etc).

Multilanguage coding is now pretty much the norm in modern enterprise development. Nearly gone are the days when a single language was used to solve all development problems.



"Wise speak: Better late than wait.

Feedback @ max@kleiner.com

Literature: Kleiner et al., Patterns konkret, 2003, Software & Support

https://github.com/maxkleiner/maXbox4/releases

https://www.academia.edu/31112544/Work_with_microservice_maXbox_starter48.pdf

https://www.php.net/manual/en/install.unix.commandline.php

https://www.piware.de/2011/01/creating-an-https-server-in-python/

Binary of CGI Exe:

https://sourceforge.net/projects/maxbox/files/Examples/13_General/880_CGI1.exe/download

1.3 References

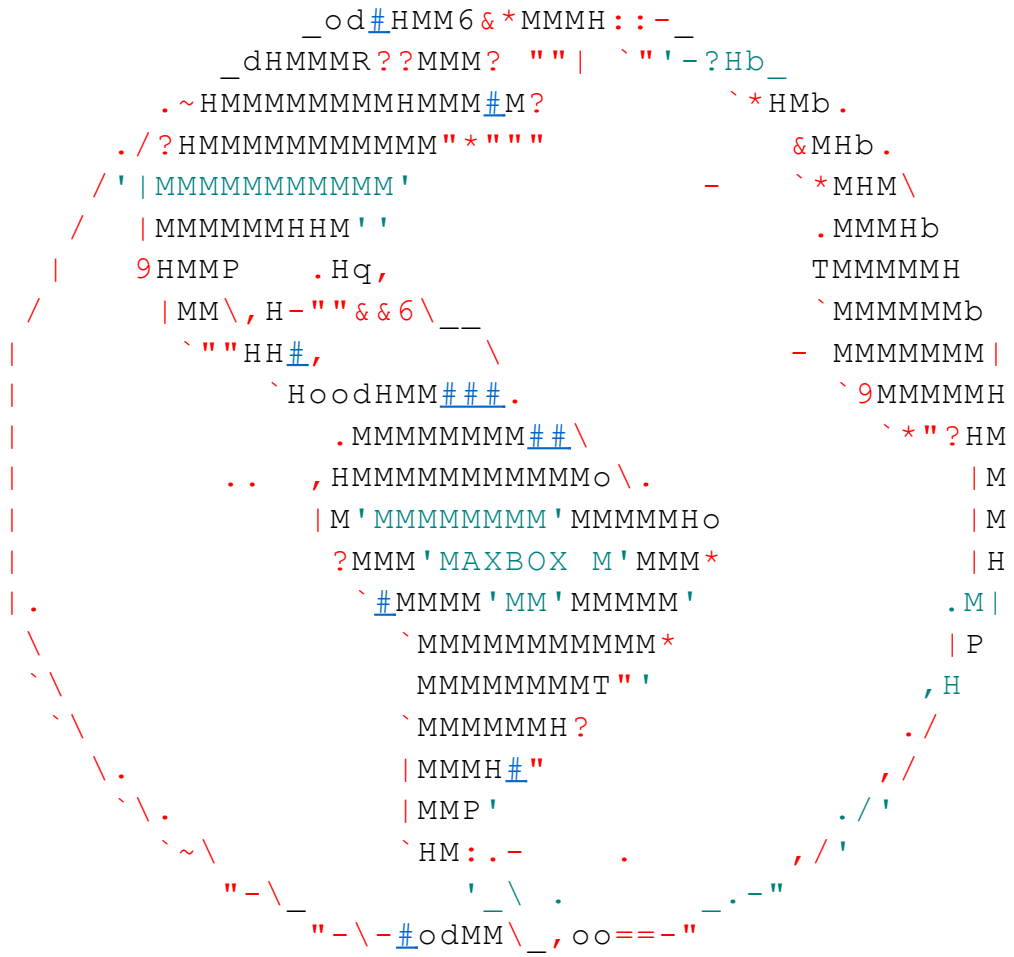
Examples and Version of this Tutorial 72:
2019-11 Cumulative Update for Windows 10 Version 1903 for x64-based Systems (KB4524570) – maXbox 4.7.1.82

https://en.wikipedia.org/wiki/WebSocket

http://www.softwareschule.ch/examples/cgi.txt

www.softwareschule.ch/examples/751_Elevator_Simulator4.pas

http://www.softwareschule.ch/examples/766_wmi_management.txt



Talk to each other across the globe, in a human voice makes a race.

https://maxbox4.wordpress.com/