Learn Light Sharp

Variables

A variable in LS (Light Sharp) is of type `any’ by default:

```
Foo = 1
```

But you can specify the type:

```
int Foo = 1
```

Functions

Calling a function in Light Sharp is like any other language:

```
Callee()
```

Defining a function is made like this:

```
def Foo(param1, param2, …):
  code;
end
```

Indeed, functions are not typed, and you can’t specify the type, you need to research what does the function returns (will be added soon)

Default functions:

```
return() “returns a value”
int() “calls a software interrupt”
syscall() “calls a syscall”
nop() “does nothing”
label() “creates a label”
goto() “jumps to a label”
```

Start function:

```
def start():
  code;
end
```

Conditions

Conditions are the most difficult part of Light Script, in order to have a condition statement, we need to:

```
$name$ = Foo == 1 “define a condition”
```

And finally,

```
$name$ callee()

$name
  callee()
$
```

Use the condition with two different syntax: multiline and inline

Imports

There are two imports:

```
with “file”
```

Used for precompiled libs (.pclslib files)

```
import “file”
```

Used for code files (.ls and .lsh files)

And, an honorable mention to

```
extern foo()
extern foovar
```

Types

```
any
int
float
ptr
string
```