

### Tipos de datos básicos, vectores y matrices.

```

int a = 0;
float time = 0.0f;
float4 pos = float4(0,0,2,1);
pos.x    // = 0
pos.z    // = 2
float2 temp = pos.yw // = float2(0,1)
pos.rgb   // = float3(0,0,2)
pos.rgb4 // = float4(0,0,2,1)
pos.aaaa // = float4(1,1,1,1)

float2x2 fMatrix = { 0.0f, 0.1, // row 1
                     2.1f, 2.2f // row 2
                   };

float4x4 mWorldViewProj; // Matriz de 4 x 4
float4x4 worldMatrix = float4( {0,0,0,0}, {1,1,1,1}, {2,2,2,2}, {3,3,3,3} );
worldMatrix[0][0]; // 1er elemento

```

### Ejemplo de Vertex Shader y estructura de vértice

```

struct VS_OUTPUT
{
    float4 Position: POSITION; // vertex position
    float4 Diffuse : COLOR0;   // vertex diffuse color
};

VS_OUTPUT Vertex_Shader_Transform(
    in float4 vPosition : POSITION,
    in float4 vColor : COLOR0 )
{
    VS_OUTPUT Output;
    Output.Position = mul( vPosition,
                           mWorldViewProj );
    Output.Diffuse = vColor;
    return Output;
}

```

### Ejemplo de Pixel shader

```

float4 Pixel_Shader( VS_OUTPUT in ) : COLOR0
{
    float4 color = in.Color;
    return color;
}

```

### Semanticas Vertex shaders

```

struct VS_OUTPUT
{
    float4 Position      : POSITION;
    float3 Diffuse       : COLOR0;
    float3 Specular      : COLOR1;
    float3 HalfVector    : TEXCOORD3;
    float3 Fresnel        : TEXCOORD2;
    float3 Reflection     : TEXCOORD0;
    float3 NoiseCoord    : TEXCOORD1;
};

```

### Inicializacion de samplers

```

sampler s = sampler_state
{
    texture = NULL;
    mipfilter = LINEAR;
};

Texture lookup
texture tex0;
sampler2D s_2D;
float2 sample_2D(float2 tex : TEXCOORD0) : COLOR
{
    return tex2D(s_2D, tex);
}

```

### Textura cúbica

```

texture tex0;
samplerCUBE s_CUBE;
float3 sample_CUBE(float3 tex : TEXCOORD0) : COLOR
{
    return texCUBE(s_CUBE, tex);
}

```

### Funciones intrínsecas HLSL

abs(value a)	absolute value (per component).
acos(x)	arccosine of each component of x.
atan(x)	arctangent of x.
atan2(y, x)	arctangent of y/x.
ceil(x)	smallest integer which is greater than or equal to x.
clamp(x, min, max)	Clamps x to the range [min, max].
cos(x)	cosine of x.
cross(a, b)	cross product of two 3D vectors a and b.
distance(a, b)	distance between two points, a and b.
dot(a, b)	dot product of two vectors, a and b.
floor(x)	greatest integer which is less than or equal to x.
fmod(a, b)	floating point remainder f of a / b
frac(x)	fractional part f of x
length(v)	length of the vector v.
lerp(a, b, s)	Returns a + s(b - a)
log(x)	base-e logarithm of x.
mul(a, b)	Performs matrix multiplication between a and b.
pow(x, y)	x^y.
reflect(i, n)	Returns the reflection vector
refract(i, n, R)	Returns the refraction vector v
round(x)	Rounds x to the nearest integer
saturate(x)	Clamps x to the range [0, 1]
sign(x)	Computes the sign of x.
sin(x)	sine of x
tan(x)	tangent of x
tex2D(s, t)	2D texture lookup.
tex2Dlod(s, t)	2D texture lookup with LOD.
transpose(m)	transpose of the matrix m.

### Texturas