1. 排版，布局和坐标系
   1. 四种布局对象（Layout Object）：
      1. Canvas 画布
      2. Section 段落
      3. Line 行
      4. Char 单个字符
   2. 编辑模式的排版（像素坐标系）

客户端原生实现了字效选择，效果配置选择（包括字体颜色，字号，分行，画布上的位置移动），将画布坐标系上的像素坐标值传入，像素坐标系如下图示意：

Canvas

Section

char

Line

* 1. PosttoText的排版（逻辑坐标系）

进入PosttoText之后，排版和布局被转换为基于opengl场景的逻辑坐标系，同时每层布局对象由基于Canvas坐标系改为基于上层布局对象的相对坐标系，逻辑坐标系如下图示意：

**Canvas高宽比**

Canvas

**Section高宽比**



Section

**1.0**



char

Line

**-1.0**



**-1.0**

**1.0**



**-Section高宽比**



**-Canvas高宽比**

Canvas坐标系中Canvas的大小为-1.0至1.0，-高宽比至高宽比；Section坐标系的原点基于Canvas坐标系描述，Section的位置和大小被描述为section坐标系原点相对于canvas坐标系原点的偏移（translate）， section的大小基于canvas大小的缩放（scale）；同理，Line基于Section坐标系，char基于Line坐标系。

因此所有布局对象基于canvas坐标系的坐标可以通过顺次叠加其上层对象的偏移和缩放得到，偏移和缩放被表达为4\*4 matrix，偏移和缩放的叠加表达为matrix的叉乘。

每种布局对象之上定义若干渲染对象，渲染对象基于对应的布局对象坐标系描述，渲染对象的渲染顺序由该对象的zorder决定，zorder大的会覆盖zorder小的。

1. 变量和表达式
   1. 效果定义中可以使用表达式定义各种数值属性，依据对象种类和依赖的布局对象不同，表达式中可以引用的变量值有所有不同；
   2. 表达式支持数值表达式和逻辑表达式；
   3. 表达式支持条件分支；
2. 资源

资源是对象引用的各种资源，目前包括如下：

* 1. Shader和Program

Shader定义着色器的代码片段；Program对应渲染程序，由若干shader组成。

* 1. Texture

Texture对应渲染时使用的纹理素材，字模也是一种纹理。

* 1. Color

Color对应某个颜色值。

* 1. Curve

Curve对应一段曲线，用于动画的速度控制。

1. 渲染对象（render object）

渲染对象是最终对界面可见的对象，渲染对象的位置被定义于某个布局对象的坐标系之中，并且其位置可以被动态改变；渲染对象的渲染方式由它使用的资源决定，使用特定的Program渲染特定纹理和颜色使其最终可见。

1. 动画

布局对象和渲染对象上可以定义若干动画，连续的，动态的改变对象的若干属性，使可见的渲染结果发生变化；动画包含两大类，值动画和复合动画：

* 1. 值动画(value animation)

值动画改变对象对应的属性值；

值动画将对象的属性在指定时间内从当前值逐渐改变到目标值；

动画执行中的某一时刻的值为开始值和目标值的差值，差值由定义的曲线决定；

动画可以循环执行；

* 1. 复合动画(composite animation)

复合动画按一定规则组合若干动画的执行，可嵌套；包括同时开始所有子动画的并行（parallel）和顺序执行子动画的串行（serial）；

1. 字效

综合以上，对字效的定义归结为：

* 1. 于合适的布局对象之上添加渲染对象；
  2. 定义渲染对象引用的资源；
  3. 定义布局对象和渲染对象的各种属性的初始值（静态）；
  4. 在各对象上定义动画改变各属性值（动态）；