# 4.Render tree construction

# Render Tree and Renderer

1. **What is Render Tree?**

   This tree is of visual elements in the **order** in which they will be displayed. It is the visual representation of the document. The purpose of this tree is to enable painting the contents in their correct order.

2. **definition of the Renderer**

   — Firefox calls the elements in the render tree "frames". WebKit uses the term renderer or render object.

   — A renderer knows how to lay out and paint itself and its children.

3. **How does renderer work?**

   — Each renderer represents a rectangular area usually corresponding to a node's CSS box. It includes geometric(几何的) information like width, height and position.

   — The box type is affected by the "display" value of the style attribute that is relevant to the node.

   — The element type is also considered: for example, form controls and tables have special frames.

DEFINITION OF WEBKIT'S RENDEROBJECT:

WebKit's RenderObject class

```
class RenderObject{
  virtual void layout();
  virtual void paint(PaintInfo);
  virtual void rect repaintRect();
  Node* node;  //the DOM node
  RenderStyle* style;  // the computed style
  RenderLayer* containgLayer; //the containing z-index layer
}
```

CREATE NODE DEPEND ON DISPLAY ATTRIBUTE:

what type of renderer should be create

```
RenderObject* RenderObject::createObject(Node* node, RenderStyle* style)
{
    Document* doc = node->document();
    RenderArena* arena = doc->renderArena();
    ...
    RenderObject* o = 0;
    switch (style->display()) {
        case NONE:
            break;
        case INLINE:
            o = new (arena) RenderInline(node);
            break;
        case BLOCK:
            o = new (arena) RenderBlock(node);
            break;
        case INLINE_BLOCK:
            o = new (arena) RenderBlock(node);
            break;
        case LIST_ITEM:
            o = new (arena) RenderListItem(node);
            break;
        ...
    }
    return o;
}
```