**对象：**

1. **DrawFigures**：该对象里包括了许多自定义绘图使用的方法，如createText、createLine等等，用户可以通过自定义绘图功能为最终生成的图形中添加额外的内容，以满足特定的需要。
2. **GraphType**：该属性保存了本次绘图的类型。
3. **ID**：该属性保存了本次绘图使用DChart产生的随机ID，用以辨识。
4. **Language**：该属性用于保存语言设置。
5. **calculatedBasic**：保存基本设置通过计算后的值。用于规范Canvas的大小及margin。
6. **canvas**：在母DIV内创建的canvas节点DOM对象。
7. **coordinates**：用于保存实时绘图坐标信息。
8. **ctx**：通过调用canvas.getContext('2d')获取的画板对象。
9. **innerData**：保存用户传入的数据。
10. **innerOptions**：保存用户传入的选项。
11. **originalDefaultOptions**：默认公共选项。每个图形类型js中，都会通过this.innerOptions = DChart.Methods.Extend(this.originalDefaultOptions, {})的方式来扩充选项。
12. **randoms**：保存DChart内部定义的用于各个对象的随机ID
13. **shapes**：保存各个图形类别各自绘图中产生的图形对象信息（用于鼠标指向时重绘、获取图形信息及鼠标点击等等操作）。
14. **tempData**：保存DChart内部类似于“通信”功能的对象，这些对象不仅在core中用到，也在特定图形类别js中用到；获取在特定图形类别中定义，但是需要传到core中使用的变量。

**事件**：

1. **onStart**：内置事件，当DChart开始绘图工作时触发的事件。
2. **onBeforeAnimation**：内置事件，当DChart准备动画前触发的事件。
3. **onAnimation**：内置事件，当DChart在动画过程中触发的事件。该事件传入参数percentAnimComplete(完成进度)。
4. **onFinish**：内置事件，当DChart完成本次绘制任务后触发的事件。

**方法**：

1. **ClearBackGround**：清空整个Canvas，包括canvas画板、为支持tip的span，保存的shapes对象等等。
2. **Draw：**关键方法，开始绘图。该方法传入两个参数：data和options。用于可同时传入两个参数，可以自传入其中一个，或者不传入任何参数。注意：调用该方法时，当data不被传入，必须保证事先已经调用SetData方法；当options不被传入，必须保证事先已经调用SetOptions方法。
3. **GetCoordinate**：获取坐标信息，参数location为需要获取坐标信息的路径(path)的字符串表达式，以"."表示子集，如"canvas.height"；如果包含数组元素，则直接以“.1”的形式获取，如以“pie.outerlabels.3.left”获取第四个半圆的OuterLabel的left数据。
4. **Initial**：重新初始化Canvas节点。包括删除旧的canvas节点(如果有的话)、删除旧的DChart样式、添加框架自定义的css样式、计算canvas的高度、宽度等基本信息、添加新的canvas节点、初始化配置信息、清空画板等等操作。总而言之，该方法目的就是初始化一个新的canvas节点，以供接下来的绘图使用。
5. **SavePic**：导出图片(部分浏览器不支持)。传入参数filename(文件名), type(文件类型，png、bmp、gif等等)
6. **SetBasicOptions**：更改基本设置。该设置可以更改画布的大小，因此当调用该方法时，其内部将调用Initial方法重新初始化。
7. **SetData**：加载数据；该方法是独立于Draw的专门加载数据的方法，其效果与在Draw中传入数据一致。
8. **SetDefaultOptions**：还原默认设置，包括公共选项以及适用于特定图形的特定选项。
9. **SetOptions**：加载选项；作用于SetData类似。
10. **SetSkin**：设定皮肤。若需保证常规选项的优先级高于皮肤设定选项的优先级，则调用SetSkin必须在SetOptions或Draw之前。
11. **\_calculateOutersValid**：收集外围要素（标题、图例、坐标等）是否绘制的信息，用于在系统中使用。
12. **\_changeTip**：更改tip的位置。当tip的位置不合适（如遮住了图形、超出边界等），需要重新对其位置进行调整。
13. **\_checkOptions**：验证所有选项输入是否正确，比如要求输入数字的选项不能输入字符串等等。相当于系统内对用户输入选项的一道关口。
14. **\_clearDrawable**：清除可绘图区域。与ClearBackGround不同的是，本方法仅清空画布(不删除其他如tip对象等内容)，且被清除的画布只是用于动画的那部分，即除去标题、图例、两个坐标轴等外围区域后剩下的长方形区域。
15. **\_clearTips：**清除所有tip的Dom对象。
16. **\_computeAxis：**计算坐标轴相关的数据信息。以供绘制坐标轴、背景比例尺及其他元素。
17. **\_computeBasic：**获取计算后的画布基本设置。若调用SetBasicOptions，则会调用本方法重新计算画布基本设置。
18. **\_computeLegend：**计算图例相关的数据信息。
19. **\_computeTitle：**计算主标题与副标题相关的数据信息。
20. **\_createAxis：**绘制坐标轴。
21. **\_createBackground：**绘制背景。包括边框及背景纯色或渐变色。
22. **\_createLegend：**绘制图例。
23. **\_createScales：**绘制背景比例尺。
24. **\_createTip：**绘制单个tip。
25. **\_createTitle：**绘制主标题和副标题。
26. **\_createAssists：**绘制除比例尺及正式图形元素外的“外围”图形。该方法用于一次性调用除动画区域外的其他区域内容，包括标题、图例、坐标轴。
27. **\_formatAxisData：**规范化传入的数据。本方法仅规范需绘制坐标轴的图形，如Area、Bar、HeapBar、HeapHistogram、Line、Histogram、Points，因此这类图形的数据结构比较复杂，而像Pie等类型的数据结构较简单，无需专门统一的规范方法。
28. **\_getCheckOptions：**获取特定图形的选项检查的配置数据。该方法定义在各个图形js中，有core在\_checkOptions中统一调用。
29. **\_getDrawableCoordinate：**获取可绘图区域的信息。用于动画中单次执行中擦除可绘制区域，以达到动画效果。
30. **\_getFormatDiff：**根据系统内的数据类型，获取两个数值之间的差值。参数有三个valueType, small, big，valueType为d时，返回small与big相差的天数；valueType为t时，返回二者相差的分钟数。
31. **\_getMouseLoction：**获取鼠标相对于Canvas的位置，返回包括X(left)和Y(top)的对象。
32. **\_onStart：**在各个绘图类型的Draw方法中调用，用于初始化或重新加载某些属性等等的“基础性工作”。在本方法中触发onStart事件。
33. **\_resetSharedOpions：**还原默认公共设置。与SetDefaultOptions不同的是，本方法仅还原公共部分的选项，而SetDefaultOptions则还原所有的选项，包括公共选项和各图形的特定选项。
34. **\_spreadSkin：**延伸皮肤设定。SetSkin方法定义在Core中，因此在各图形js中各自定义\_spreadSkin，然后统一由core的SetSkin调用，以达到各个图形都能定义自身特定皮肤选项的目的。
35. **\_startDrawAndAnimation：**开始绘图。定义在core，由各图形js在Draw方法中调用。
36. **\_getComputed**：统一获取线值差、最大最小值、线条数目的算法。该方法通过传入的数据类型、实际最大最小值等信息，计算出参考最大最小值、参考值差(Interval)及“值段的数目”(即从最小值累加值差到最大值的过程中，累加的次数)。