

Lorem ipsum dolor sit amet, consectetur adipiscing elit
 sed do eiusmod tempor incididunt ut labore et dolore
 magna aliqua. Ut enim ad minim veniam, quis nostrud
 exercitation ullamco laboris nisi ut aliquip ex ea
 commodo consequat. Duis aute irure dolor in reprehensi-
 dent in voluptate velit esse cillum dolore eu fugiat nulla
 pariatur.

EPS 10 ABSTRACT GRAPHIC
vector Illustration

可视化创建小程序 方案分享

咫尺网络 - 安乐

如何可视化创建小程序？

- | | |
|------------|-------------|
| 1、可视化操作编辑器 | • 创建应用、保存数据 |
| 2、小程序代码包生成 | • 分析数据、生成代码 |
| 3、应用管理后台 | • 查看数据、管理内容 |
| 4、web端效果预览 | • 无需打包、提前预览 |



可视化操作编辑器

页面管理

组件库

量尺

125%

我的应用52

版本历史

提交为模块

后台管理

预览

保存

发布

文本

图片

按钮

标题

高级列表

视频

轮播

公告

添加组合

布局

功能

营销

其他

双栏

分割线

弹窗

面板

自由面板

滑动面板

动态分类

侧边栏

悬浮窗

前进

后退

模板

元素

添加导航

首页

即速应用

图片组件

组件样式

组件类

单图片

图片列表

图文集

更多模板

上传图片

背景

图片

纯色

更换图片

透明度100%

点击事件

设置点击事件

编辑器是小程序数据的来源

小程序开发文档

- 框架：目录结构，全局配置、页面注册、语法、基础能力、开放能力、插件
- 组件：组件拼装组成页面
- API：封装成点击事件供用户设置

组件

用户使用组件组建页面

- 基础组件
- 容器组件
- 功能组件
- 组件模板

事件

实现页面间、组件间的交互逻辑




- 跳转页面
- 组件交互
- 小程序API

其他

方便用户操作，提供更多功能

- 数据对象：健身房
- 拖拽、排序、复制、回退
- 动画
- 页面设置
- 恢复历史数据、更改主题等

每个组件包含 **model**、**view**、**setting** 三部分

 textSetting	2018/11/13 9:47	文件夹	
 textView	2018/10/9 17:19	文件夹	
 text.model.ts	2018/10/22 17:20	TS 文件	2 KB

- 1、text.model.ts 组件数据
- 2、textView 组件模板
- 3、textSetting 组件设置区

文本组件model, 方法继承自Widget, 组件保存时存储content、customFeature、style字段

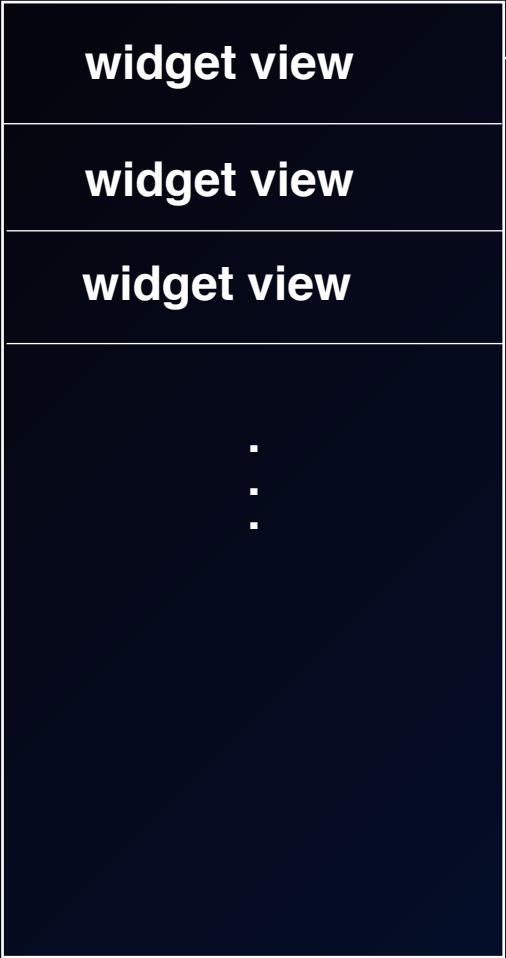
```
import { Widget } from '../widget.model';

chenchao, 2 months ago | 3 authors (You and others)
export class Text extends Widget {
  type:string = 'text';
  name:string = "文本";

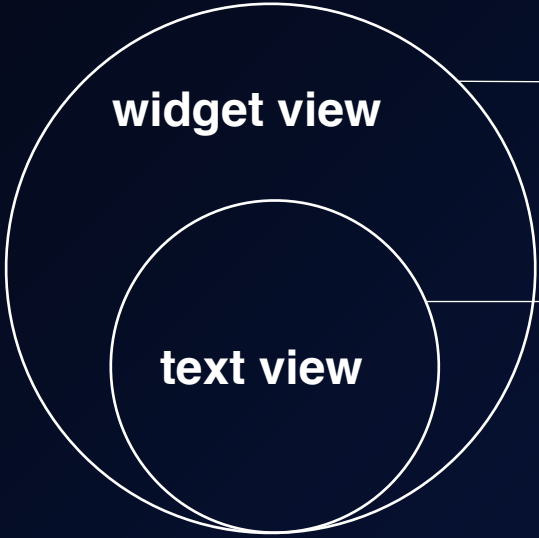
  constructor(config?:any){
    super(config);

    this.content || (this.content = '我是文本');
    this.customFeature || (this.customFeature = {
      boxColor:"rgb(0, 0, 0)",
      boxR:"5",
      boxStyle:false,
      boxX:"0",
      boxY:"0",
      textColor:"rgb(0, 0, 0)",
      textStyle:false,
      textX:"0",
      textY:"0",
      textR:"5",
      isWordWrap: 0,
      dataObject: false,
      'word-wrap': 2,
      phoneNumberSource: 'static',
      phoneDisplayContent: 'static',
      name:'文本',
      isLockWidget :false
    });
    this.style || (this.style = { ...
  });
}
}
```


`widgetList : Widget<array>`



model



动态组件

删除、拖拽、缩放等公共功能

组件样式和内容

widget view

组件外壳

```
<div class="control-wrap">
  <span class="move-handle"><i></i><i></i><i></i></span>
  <span (click)="removeWidget($event)" class="ele-delete-icon" title="删除"><x</span>
</div>
<template #placeholder></template>
<div class="curchange-border-decoration">
  <div></div><div></div><div></div><div></div>
  <span></span><span></span><span></span><span></span><span></span><span></span><span></span><span></span><span></span><span></span>
</div>
```

text view

组件内容

```
<div *ngIf="widget?.pattern === 'free'" class="ele-content element {{widget?.type}}" [ngStyle]="{
  'background-color': widget.style['background-color'],
  'background-image': widget.style['background-image'],
  'border-style': widget.style['border-style'],
  'border-color': widget.style['border-color'],
  'border-width': widget.style['border-width'],
  'box-shadow': widget.style['box-shadow']
}">
  <div class="inner-content {{ widget.customFeature.isWordWrap == 1 ? 'text-word-wrap' : '' }}" [ngStyle]="{ ...
  }">{{ widget.content }}</div>
</div>
<div *ngIf="widget?.pattern !== 'free'" class="ele-content element {{widget?.type}}" [ngStyle]="{
  width: widget.style.width,
  'margin-top': widget.style['margin-top'],
  'margin-left': widget.style['margin-left'],
  'margin-right': widget.style['margin-right'],
  'background-color': widget.style['background-color'],
  'background-image': widget.style['background-image'],
  'border-style': widget.style['border-style'],
  'border-color': widget.style['border-color'],
  'border-width': widget.style['border-width'],
  'box-shadow': widget.style['box-shadow']
}">
  <div class="inner-content {{ widget.customFeature.isWordWrap == 1 ? 'text-word-wrap' : '' }}" [ngStyle]="{ ...
  }">{{ widget.content }}</div>
</div>
```

组件

用户使用组件组建页面

- 基础组件
- 容器组件
- 功能组件
- 组件模板

事件

实现页面间、组件间的交互逻辑

- 跳转页面
- 组件交互
- 小程序API

其他

方便用户操作，提供更多功能

- 数据对象：健身房
- 拖拽、排序、复制、回退
- 动画
- 页面设置
- 恢复历史数据、更改主题等

组件

用户使用组件组建页面

- 基础组件
- 容器组件
- 功能组件
- 组件模板

事件

实现页面间、组件间的交互逻辑

- 跳转页面
- 组件交互
- 小程序API

其他

方便用户操作，提供更多功能

- 数据对象：健身房
- 拖拽、排序、复制、回退
- 动画
- 页面设置
- 恢复历史数据、更改主题等





小程序代码包生成

wxml

组件模板组合而成
的页面结构

wxss

组件样式
页面样式

js

初始数据
组件事件处理函数
生命周期回调函数
页面事件处理函数

json

页面配置信息

wxml

不同组件模板
按照顺序
排列、嵌套组成
的页面结构



组件模板

- template、自定义组件
- 组件数据
- 点击事件



排列嵌套

- 组件种类多
- 嵌套关系复杂

数据获取方式 ①

组件的位置 ②



③ 展示内容的类型

```
<view class="ele-container" wx:if="{{!{$compId/}.hidden}}">
  <block wx:if="{{{$compId/}.customFeature.segment == 'default_map' && real_value}}">
    <!-- 在动态列表中的双栏中绑定地图 -->
    <template is="zcText" data="{{data:{$compId/}, real_value:real_value.form_data[{$compId/}.customFeature.segment], form_data:
      real_value.form_data}}"></template>
  </block>
  <block wx:elif="{{{$compId/}.customFeature.segment == 'default_map'}}">
    <!-- 动态列表, 动态容器 与详情页绑定地图 -->
    <template is="zcText" data="{{data:{$compId/}, real_value:item.form_data[{$compId/}.customFeature.segment], parent_form:item.form,
      form_data: item.form_data, detail_data: detail_data}}"></template>
  </block>
  <block wx:elif="{{real_value}}">
    <template is="zcText" data="{{data:{$compId/}, real_value:real_value.form_data[{$compId/}.customFeature.segment], parent_form:item.f
      detail_data: detail_data}}"></template>
  </block>
  <block wx:elif="{{detail_data[{$compId/}.customFeature.segment] && in_dynamic_vessel !== 'true'}}">
    <template is="zcText" data="{{data:{$compId/}, real_value:detail_data[{$compId/}.customFeature.segment], parent_form:item.form,
      detail_data: detail_data}}"></template>
  </block>
  <block wx:elif="{{in_dynamic_vessel === 'true'}}">
    <template is="zcText" data="{{data:{$compId/}, real_value:item.form_data[{$compId/}.customFeature.segment], parent_form:item.form,
      detail_data: detail_data}}"></template>
  </block>
  <block wx:else>
    <template is="zcText" data="{{data:{$compId/}, real_value:item.form_data[{$compId/}.customFeature.segment], parent_form:item.form,
      detail_data: detail_data}}"></template>
  </block>
</view>
```

点击事件

不同点击事件需要不同的模板结构：跳转小程序

```
<template name="zcText">
  <block wx:if="{{real_value && (data.customFeature.action !== 'call' || data.customFeature.phoneDisplayContent !== 'static')}}">
    <block wx:if="{{textWxsModule.isArray(real_value)}}">
      <block wx:if="{{data.canIUseNavigatorTarget && data.customFeature.action == 'turn-to-xcx'}}">
        <navigator target="miniProgram" open-type="navigate" app-id="{{data.customFeature['xcx-appid']}}" path="{{data.customFeature['xcx-page-url'] || ''}}" version="release">
          <view class="element text {{data.customFeature.isWordWrap == 1 ? 'text-word-wrap' : '' }}" style="{{data.style}}{{data.customFeature.isWordWrap == 1 ?
            '-webkit-line-clamp:' + data.customFeature['word-wrap'] : '' }}" bindanimationend="animationEnd" data-compid="{{ data.compId }}">
            <block wx:if="{{ parent_form === 'form' }}">{{ real_value }}</block>
            <block wx:else>
              <view class="wxParse"><template is="wxParse" data="{{wxParseData: real_value}}" /></view>
            </block>
          </view>
        </navigator>
      </block>
      <block wx:else>
        <view data-compdata="{{data}}" data-real-value="{{ real_value }}" class="element text {{data.customFeature.isWordWrap == 1 ? 'text-word-wrap' : '' }}" style="{{data.style}}{{
          data.customFeature.isWordWrap == 1 ? '-webkit-line-clamp:' + data.customFeature['word-wrap'] : '' }}"
          data-event-params="{{data.eventParams}}" catchtap="{{data.eventHandler}}" bindanimationend="animationEnd" data-compid="{{ data.compId }}">
            <block wx:if="{{ parent_form === 'form' }}">{{ real_value }}</block>
            <block wx:else>
              <view class="wxParse"><template is="wxParse" data="{{wxParseData: real_value}}" /></view>
            </block>
          </view>
        </block>
      </block>
    </block>
    <block wx:else>
      <block wx:if="{{data.canIUseNavigatorTarget && data.customFeature.action == 'turn-to-xcx'}}">
        <navigator target="miniProgram" open-type="navigate" app-id="{{data.customFeature['xcx-appid']}}" path="{{data.customFeature['xcx-page-url'] || ''}}" version="release">
          <text class="element text {{data.customFeature.isWordWrap == 1 ? 'text-word-wrap' : '' }}" style="{{data.style}}{{data.customFeature.isWordWrap == 1 ?
            '-webkit-line-clamp:' + data.customFeature['word-wrap'] : '' }}" bindanimationend="animationEnd" data-compid="{{ data.compId }}">{{data.customFeature.phoneDisplayContent
            === 'static' ? data.content : real_value}}</text>
          </navigator>
        </block>
        <block wx:else>
          <text data-compdata="{{ data }}" class="element text {{data.customFeature.isWordWrap == 1 ? 'text-word-wrap' : '' }}" style="{{data.style}}{{data.customFeature.isWordWrap
            == 1 ? '-webkit-line-clamp:' + data.customFeature['word-wrap'] : '' }}" data-event-params="{{data.eventParams}}"
            catchtap="{{data.eventHandler}}" bindanimationend="animationEnd" data-compid="{{ data.compId }}">{{data.customFeature.phoneDisplayContent === 'static' ? data.content :
            real_value}}</text>
          </block>
        </block>
      </block>
    </block>
  </template>
```

组件种类较多，且容器互相嵌套情况复杂，由后端处理



1. 默认样式

组件默认样式写在公共样式里

2. 用户设置样式

组件：模板内联样式覆盖

页面：生成代码包时写入

js

初始数据
组件事件处理函数
生命周期回调函数
页面事件处理函数

1. 初始数据

data字段，生成代码包时的页面数据

2. 点击事件

组件模板上调用的函数，具体代码写在公共部分里

```
var clickEventHandler = {
  "goods-trade": function(param, franchisee) {
    let gtype = param['goods-type'] || param['goods_type'];
    let id = param['goods-id'] || param['goods_id'];
    if(!id){
      return;
    }
    let queryStr = franchisee ? '&franchisee=' + franchiseeId : '';
    if (gtype == 3) {
      app.turnToPage('/pages/toStoreDetail/toStoreDetail?detail=' + id + queryStr);
    } else {
      app.turnToPage('/pages/goodsDetail/goodsDetail?detail=' + id + queryStr);
    }
  },
  "to-seckill": function (param, franchisee) { ...
},
  "inner-link": function (param, franchisee) { ...
},
  "call": function (param, franchisee) { ...
},
  "get-coupon": function (param, franchisee) { ...
},
  "community": function (param, franchisee) { ...
},
  "franchisee-enter": function(param, franchisee){ ...
},
  "to-franchisee": function (param, franchisee) { ...
},
  "to-promotion": function (param, franchisee) { ...
},
  "coupon-receive-list": function (param, franchisee) { ...
},
  "recharge": function (param, franchisee) { ...
}
```


js

初始数据
组件事件处理函数
生命周期回调函数
页面事件处理函数

3. 生命周期回调函数

onLoad: 页面数据初始化、登录状态

onShow: 刷新页面状态

onReady: 更改页面标题

onHide: 隐藏组件（弹窗、侧边栏等）

onUnload: 清除页面相关的全局变量和设置

4. 页面事件处理函数

onPullDownRefresh: 页面重载

onReachBottom: 加载列表

onPageScroll: 回到页面顶部

onShareAppMessage: 统计分享次数、分享后送积分等

onTabItemTap: 清除导航状态

小程序代码包生成—json

json

页面配置信息

生成小程序代码包时写入：下拉刷新、页面标题等

功能页面

- 电商、到店、外卖、多商家、营销类（拼团、秒杀、会员卡、优惠券、储值、大转盘、刮刮乐）
- 社区、资讯
- 插件类

功能独立、分包加载、按需打包

动态更新

效果：修改组件内容，不需要重新打包即可更新线上版本内容，减少用户打包次数。

方法：配合编辑器内的历史保存记录。



应用管理后台

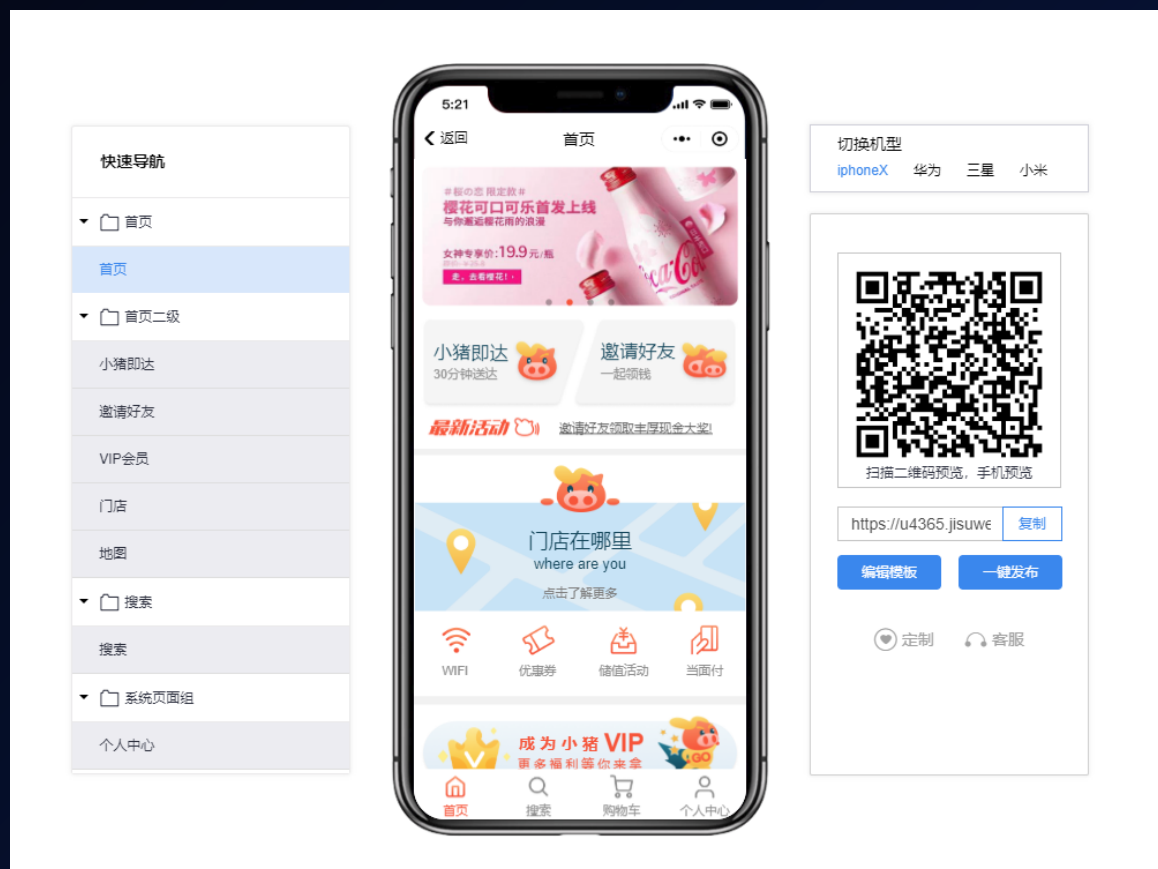
数据统计、用户管理、数据对象管理、商品数据管理、图片库、视频库等

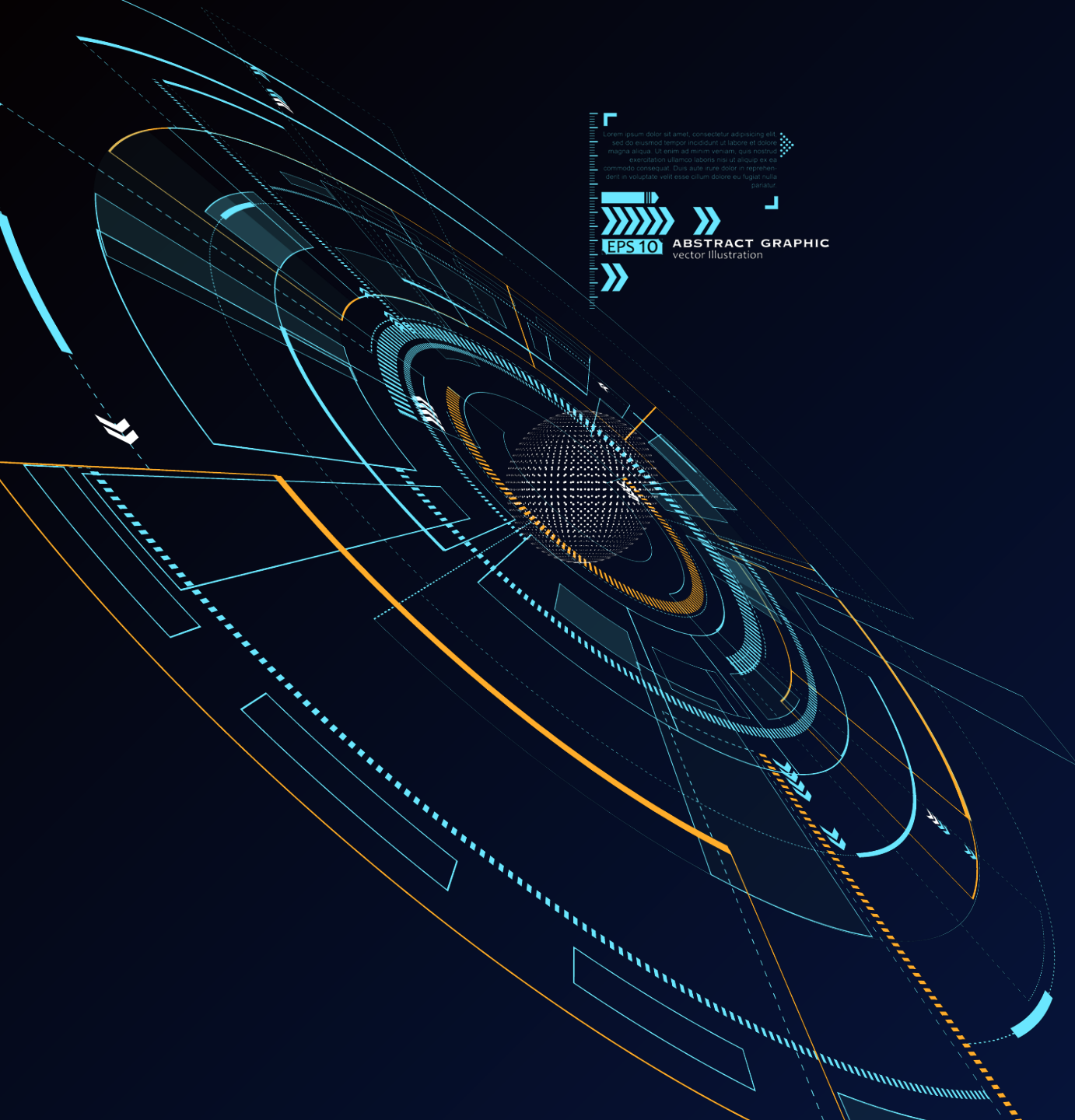




web端预览

无需打包 提前预览 & H5应用





Lorem ipsum dolor sit amet, consectetur adipiscing elit
 sed do eiusmod tempor incididunt ut labore et dolore
 magna aliqua. Ut enim ad minim veniam, quis nostrud
 exercitation ullamco laboris nisi ut aliquip ex ea
 commodo consequat. Duis aute irure dolor in reprehensi-
 dent in voluptate velit esse cillum dolore eu fugiat nulla
 pariatur

EPS 10 ABSTRACT GRAPHIC
vector Illustration

谢谢观看