



# ICP区块链开发入门课程

1. 使用 SDK 搭建第一个网站

主讲: Paul Liu - DFINITY 工程师

# 课程大纲

1. 使用 SDK 搭建一个简易网站
2. Motoko 语言简介
3. Canister 智能合约
4. 用 Motoko 做后端
5. 用 Javascript 做前端



# 不一样的区块链 ICP

ICP (全称 Internet Computer Protocol 互联网计算机协议 <https://internetcomputer.org>) 是 DFINITY 基金会 (<https://dfinity.org>) 参与孵化的去中心化的区块链项目

## 100% on-chain web3

Chain Key crypto - advanced smart contracts process HTTP, control other chains, and scale

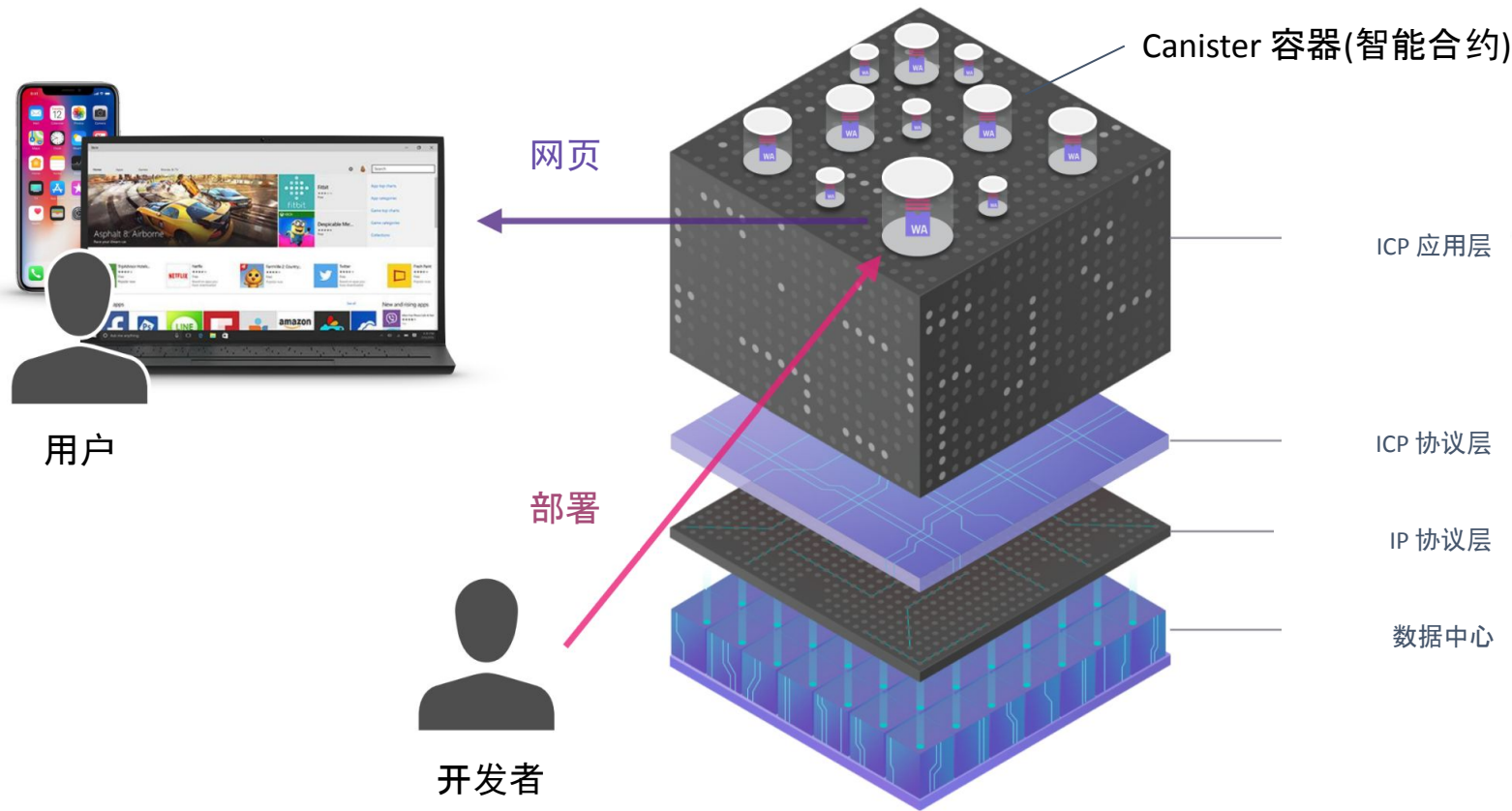
BUILD REAL WEB3

→ Explore the Internet Computer ecosystem



中文网站: <https://ic123.xyz>

# ICP 区块链的应用架构



# 安装 ICP 软件开发工具包 (SDK)

前往 <https://ic123.xyz/developcenter/dfxinstall> (中文)  
或者 <https://github.com/dfinity/sdk> (英文) 网站

## Mac/Linux安装

通过终端输入如下命令，即可安装dfx，如果下载途中提示是否继续下载，请输入y即可。

```
sh -ci "$(curl -fsSL https://smartcontracts.org/install.sh)"
```

## Windows安装使用

Dfx原生并不支持Windows系统，不过你可以安装Windows Subsystem for Linux (WSL) 来运行DFX。

- 
1. 点击此处拷贝命令行
  2. 粘贴至 Shell 终端并执行
  3. 等候安装结束

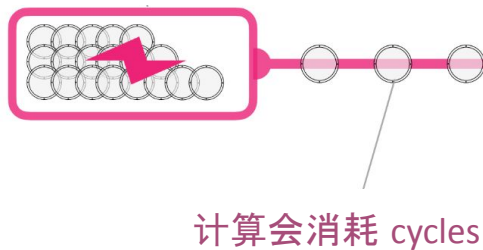




# DFX 演示

# 使用 cycles 为智能合约付费

- 运行智能合约 (Canister容器) 的成本通过 cycles 来计量
- 每个容器消耗各自的 cycles 余额
- 价格稳定  $1 \times 10^{12}$  cycles = 1 SDR (约 US\$1.41)
- 通常由开发者为自己的容器充值



开发者可以免费获取 cycles  
<https://faucet.dfinity.org>

# 用优惠码免费领取 cycles 钱包

开发者需要使用 Cycles 钱包才能在主网部署合约

用优惠码领取步骤:

1. 下载并安装 SDK
2. 使用 dfx 命令检查自己的 principal id

```
dfx identity get-principal
```

3. 使用 dfx 命令兑换 cycles 钱包

```
dfx canister --network=ic \  
    call fg7gi-vyaaa-aaaal-qadca-cai redeem '("xxxxx-yyyyy-zzzzz")'
```

4. 使用 dfx 关联钱包

```
dfx identity --network=ic set-wallet CANISTER_ID
```





# Cycles 演示

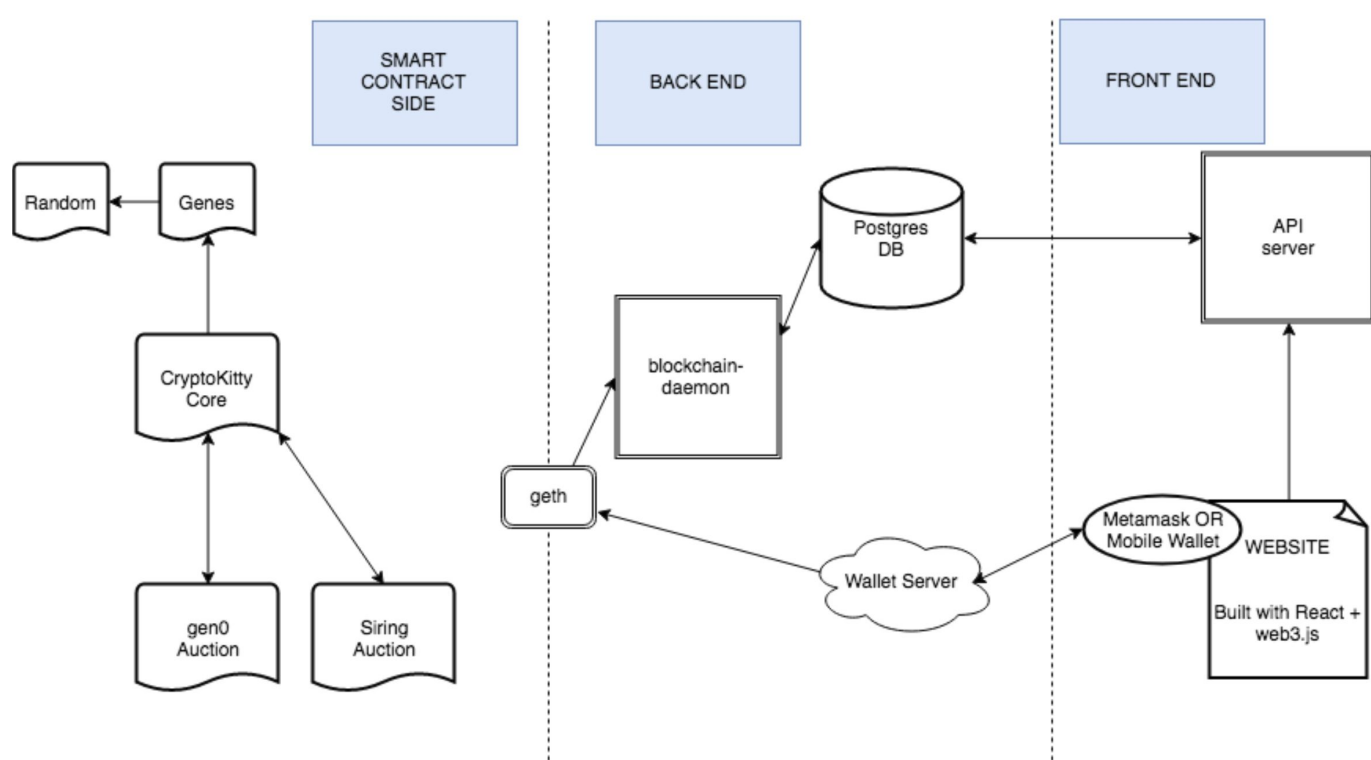


# 使用 ICP 的技术优势

去中心链上治理  
可验证无偏倚随机序列  
毫秒级查询响应 容器状态自动持久化  
无缝代码升级 消息处理原子性  
48字节ChainKey公钥验证  
状态隔离 两秒交易最终确认 弹性扩容  
异步 负载均衡 并发  
可信任计算

# 其它区块链 DAPP 架构示例

## 加密猫 (CryptoKitties) 的系统架构



# ICP 对 DAPP 架构的简化

`https://<canister-id>.ic0.app` 或者  
`https://用户域名.ic0.app`



- 轻量级容器封装代码和数据 (canister)
- 直接发送 HTML + JS 至用户终端
- 无需第三方钱包软件或插件
- JS Agent 库提供底层支持  
(远程方法调用, 密钥管理, 身份认证, 交易验证等等)



# ICP 对 WEB 架构的简化

~~Cloud Services~~

~~Firewalls~~

~~Web Servers~~

~~Middleware~~

~~Databases~~

~~Application Servers~~

~~Load Balancers~~

~~Memcached~~

~~CDNs~~

~~Username & Passwords~~

~~DNS services~~

~~Server Anti-Virus~~

# 解读“去中心化”

数据安全  
计算可信

对安全性的  
需求



去中心化

网络节点  
激励机制  
数据的权力  
代码的权力  
分布式治理

# 课程作业

1. 使用 SDK 搭建一个简易网站

```
dfx new --no-frontend
```

2. 领取 cycles 钱包
3. 将网站部署到 ic0.app 主网
4. 思考题:假如开发团队不再维护代码了  
，用户该怎么办？

## 下一节: Motoko 语言简介

- VSCode 开发环境配置
- 语法以及编程基础
- 运行 Motoko 代码
- Motoko Playground