# dfx new

dfx new用于创建一个新项目。 此命令使用模板文件创建默认项目结构。必须指定要创建的项目的名称。

## 语法

dfx new [FLAGS] [OPTIONS] <project-name>

## 选项

### FLAGS

* --dry-run：提供要创建的目录和文件的预览，但不将它们添加到文件系统。
* --frontend：是否需要要前端代码示例。如果安装了 Node，则默认为 true，否则默认为 false。
* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --agent-version <agent-version>：覆盖要安装的 JavaScript 代理版本。默认会根据 NPM 决定。

### project-name

指定要创建的项目的名称，该参数必须指定。

## 示例

1. dfx new my\_social\_network

创建一个名为 my\_social\_network 的新项目。

1. dfx new my\_social\_network --dry-run

预览要创建的目录和文件，而不将它们添加到文件系统。

# dfx start

dfx start 命令为当前项目启动本地 Internet 计算机副本进程和 Web 服务器进程。

请注意，只能从项目目录结构中运行此命令。

## 语法

dfx start [FLAGS] [OPTIONS]

## 选项

### FLAGS

* --background：在后台启动 Internet 计算机副本和 Web 服务器进程，并在返回 shell 之前等待回复。
* --clean：通过从项目缓存中删除检查点，来以干净的状态启动 Internet 计算机副本和 Web 服务器进程。
* --emulator：运行专用模拟器而不是副本。
* --no-artificial-delay：去除为了模拟IC联网环境而添加的本地副本中的人为延迟。
* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --host host：指定要绑定前端的主机接口 IP 地址和端口号。默认值为 127.0.0.1:8000。

## 示例

1. dfx start

正常启动 Internet 计算机副本和 Web 服务器进程。

1. dfx start --background

在后台启动 Internet 计算机副本和 Web 服务器进程。

# dfx bootstrap

使用 dfx bootstrap 命令启动在 dfx.json 配置文件中定义或使用命令行选项指定的Web 引导服务器。引导服务器用于为项目提供前端静态资产。

## 语法

dfx bootstrap [FLAGS] [option]

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --ip <ip>：指定引导服务器监听的 IP 地址。如果不指定，则使用dfx.json中配置的地址。 默认情况下，服务器地址为 127.0.0.1。
* --port <port>：指定引导服务器监听的端口号。 默认情况下，使用端口号 8081。
* --network <network>：如果要覆盖默认的本地网络端点 (http://127.0.0.1:8080/api)，则指定要连接的网络。
* --timeout <timeout>：指定引导服务器等待上游请求完成的最长时间（以秒为单位）。默认为30 秒。
* --root <root>：指定引导服务器服务的静态资产目录。默认：$HOME/.cache/dfinity/versions/$DFX\_VERSION/js-user-library/dist/bootstrap。

## 示例

1. dfx bootstrap --ip 192.168.47.1 --port 5353

使用特定的 IP 地址和端口号启动引导服务器。

1. dfx bootstrap --root $HOME/ic-projects/assets --timeout 60

使用默认服务器地址和端口号，但为静态资产指定自定义位置和更长的超时时间。

# dfx replica

使用 dfx replica 命令在本地启动 Internet 计算机副本进程（没有 Web 服务器）。

请注意，只能从项目目录结构中运行此命令。

## 语法

dfx replica [FLAGS] [OPTIONS]

## 选项

### FLAGS

* --emulator：运行专用模拟器而不是副本。
* --no-artificial-delay：去除为了模拟IC联网环境而添加的本地副本中的人为延迟。
* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --port <port>：指定本地副本应该侦听的端口。

## 示例

1. dfx replica

启动 Internet 计算机副本

# dfx stop

使用 dfx stop 命令停止当前在本地计算机上运行的 Internet 计算机进程。

请注意，只能从项目目录结构中运行此命令。

## 语法

dfx stop [FLAGS]

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

## 示例

1. dfx stop

停止在后台运行的 Internet 计算机网络进程。

# dfx ping

使用 dfx ping 命令检查本地计算机和 Internet 计算机网络提供商之间的连接。

请注意，只能从项目目录结构中运行此命令。

## 语法

dfx ping [FLAG] [provider]

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### provider

指定网络提供商 URL。如果未提供此参数，则默认ping本地网络。名称“IC”表示Internet Computer主网。

## 示例

1. dfx ping <https://testgw.dfinity.network>

检查 Internet 计算机当前是否在指定网络地址可用。

# dfx identity

dfx identity 命令用于管理与 Internet 计算机网络通信的身份。创建多个用户身份使您能够测试基于用户的访问控制。

## 语法

dfx identity [FLAGS] [OPTIONS] <SUBCOMMAND>

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --network <network>：指定要连接的计算网络。默认使用本地网络。“IC”表示Internet Computer主网。

## 子命令

#### whoami

使用 dfx identity whoami 命令显示当前活动的用户身份上下文的名称。

##### 语法

dfx identity whoami [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx identity whoami

显示当前活动用户身份的名称。

#### new

使用 dfx identity new 命令添加新的用户身份。这些身份的凭据存储在 $HOME/.config/dfx/identity/<identity\_name>/identity.pem 文件中。

注意：添加的身份是全局的，并不局限于特定的项目环境。

##### 语法

dfx identity new [FLAGS] [OPTIONS] <identity>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### OPTIONS

* --hsm-key-id <hsm-key-id>：一系列十六进制数字对。
* -hsm-pkcs11-lib-path <hsm-pkcs11-lib-path>：opensc-pkcs11 库的文件路径，例如 “/usr/local/lib/opensc-pkcs11.so”

###### identity

指定要创建的身份的名称。 这个参数是必需的。

##### 示例

1. dfx identity new ic\_admin

创建一个名为 ic\_admin 的身份。

#### import

使用 dfx identity import 命令通过从 PEM 文件导入用户的密钥信息或安全证书来创建用户身份。

注意：添加的身份是全局的，并不局限于特定的项目环境。

##### 语法

dfx identity import [FLAGS] <identity> <pem-file>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### identity

指定要创建的身份的名称。 这个参数是必需的。

###### pem-file

指定用于创建的身份的PEM文件。 这个参数是必需的。

##### 示例

1. dfx identity import alice generated-id.pem

导入生成的 id.pem 文件来创建用户身份 alice。

#### remove

使用 dfx identity remove 命令删除现有的用户身份。

注意：删除的身份是全局的，并不局限于特定的项目环境。

##### 语法

dfx identity remove [FLAGS] <identity>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### identity

指定要删除的身份的名称。 这个参数是必需的。

##### 示例

1. dfx identity remove default

删除默认用户身份。

#### rename

使用 dfx identity rename 命令重命名现有用户身份。

注意：重命名的身份是全局的，并不局限于特定的项目环境。

##### 语法

dfx identity rename [FLAGS] <from> <to>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### from

指定要重命名的身份的名称。 这个参数是必需的。

###### to

指定新的身份的名称。 这个参数是必需的。

##### 示例

1. dfx identity rename test\_admin devops

将test\_admin身份重命名为devops。

#### use

使用 dfx identity use 命令指定要激活的用户身份。

注意：激活的身份是全局的，并不局限于特定的项目环境。

##### 语法

dfx identity use [FLAGS] <identity>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### identity

指定要激活的身份的名称。 这个参数是必需的。

##### 示例

1. dfx identity use ops

激活ops身份。

#### list

使用 dfx identity list 命令显示可用的用户身份列表。运行此命令时，列表会显示一个星号 (\*) 以指示当前活动的用户上下文。

##### 语法

dfx identity list [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx identity list

显示所有身份。

#### get-principal

使用 dfx identity get-principal 命令显示与当前用户身份上下文关联的主体的文本表示。这是一串字符。

如果您尚未创建任何用户身份，则可以使用此命令显示默认用户的主体。

##### 语法

dfx identity get-principal [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx identity get-principal

显示当前用户身份关联的主体的文本表示。

#### deploy-wallet

将钱包 WASM 安装到指定的canister。该canister必须未安装过程序，且其controller为当前身份。

注意：必须连接到 Internet 计算机网络或在本地运行网络才能运行此命令。此外，必须在项目目录中才能运行该命令。

##### 语法

dfx identity deploy-wallet [FLAGS] <canister-id>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### canister-id

部署钱包 WASM 的canister ID，该canister必须未安装过程序，且其controller为当前身份。

##### 示例

1. dfx identity --network ic deploy-wallet gastn-uqaaa-aaaae-aaafq-cai

向IC主网的ID为gastn-uqaaa-aaaae-aaafq-cai的canister安装wallet WASM。

#### get-wallet

使用 dfx identity get-wallet 命令显示与当前身份主体关联的钱包canister ID。

注意：必须连接到 Internet 计算机网络或在本地运行网络才能运行此命令。此外，必须在项目目录中才能运行该命令。

##### 语法

dfx identity get-wallet [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx identity –network ic get-wallet

显示与IC主网上当前身份主体关联的钱包canister ID。

#### set-wallet

使用 dfx identity set-wallet 命令将钱包canister ID与当前身份进行关联。

##### 语法

dfx identity set-wallet [FLAGS] <canister-name>

##### 选项

###### FLAGS

* --force：跳过验证指定的canister是有效的钱包canister。
* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx identity –network ic set-wallet –force gastn-uqaaa-aaaae-aaafq-cai

强制将IC主网上的钱包canister gastn-uqaaa-aaaae-aaafq-cai与当前身份进行关联。

# dfx ledger

使用 dfx ledger 命令与ledger canister交互。

此命令可用于进行 ICP令牌交易，或使用ICP 对canister充值cycle。

## 语法

dfx ledger [FLAGS] [OPTIONS] <SUBCOMMAND>

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --network <network>：指定要连接的计算网络。默认使用本地网络。“IC”表示Internet Computer主网。

## 子命令

#### create-canister

使用 dfx ledger create-canister 命令将 ICP 令牌转换为cycle并在 Internet 计算机上创建新的canister。

##### 语法

dfx ledger create-canister [FLAGS] [OPTIONS] <controller>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### OPTIONS

* --amount <amount>：指定要将多少ICP铸造为cycle，这些cycle将存入canister。金额可以指定为最多八位小数的数字。
* --e8s <e8s>：指定ICP 令牌的小数单位。可以单独使用此选项或与 --e8s 结合使用。--e8s 5000000表示0.05000000个ICP。
* --icp <icp>：指定ICP 令牌的整数单位。可以单独使用此选项或与 --e8s 结合使用。
* --fee <fee>：指定交易费用。 默认值为 10000 e8s。即0.0001个ICP。
* --max-fee <max-fee>：指定最高交易费用。 默认值为 10000 e8s。即0.0001个ICP。

###### controller

指定要设置为新canister的控制器的身份主体标识符。必须要设置此参数。controller决定了谁可以操作此canister。

##### 示例

1. dfx ledger --network ic create-canister tsqwz-udeik-5migd-ehrev-pvoqv-szx2g-akh5s-fkyqc-zy6q7-snav6-uqe --amount 1.25

创建一个新的canister，1.25个ICP将转换为cycle并存入新的canister。

1. dfx ledger --network ic create-canister tsqwz-udeik-5migd-ehrev-pvoqv-szx2g-akh5s-fkyqc-zy6q7-snav6-uqe --icp 3 --e8s 5000

创建一个新的canister，3.0.00005个ICP将转换为cycle并存入新的canister。

#### account-id

使用 dfx ledger account-id 命令显示与当前活动身份关联的ledger帐户标识符(ICP地址)。

与开发人员身份主体的文本表示一样，ledger帐户标识符源自私钥，用于在ledger canister中表示身份。

##### 语法

dfx ledger account-id [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx ledger --network ic account-id

获取IC主网上当前活动身份关联的ledger帐户标识符。

#### balance

使用 dfx ledger balance 命令打印您的帐户余额或其他用户的余额。

##### 语法

dfx ledger balance [FLAGS] [of]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### of

指定帐户标识符以获取余额。 如果未指定该命令，则该命令返回当前选择的用户身份的 ICP 代币余额。

##### 示例

1. dfx ledger --network ic balance 03e3d86f29a069c6f2c5c48e01bc084e4ea18ad02b0eec8fccadf4487183c223

获取指定账户的ICP余额。

#### transfer

使用 dfx ledger transfer 命令将 ICP 代币从您的帐户地址转移到目标地址。

##### 语法

dfx ledger transfer [FLAGS] [OPTIONS] <to>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### OPTIONS

* --amount <amount>：指定要发送多少ICP。金额可以指定为最多八位小数的数字。
* --e8s <e8s>：指定ICP 令牌的小数单位。可以单独使用此选项或与 --e8s 结合使用。--e8s 5000000表示0.05000000个ICP。
* --icp <icp>：指定ICP 令牌的整数单位。可以单独使用此选项或与 --e8s 结合使用。
* --fee <fee>：指定交易费用。 默认值为 10000 e8s。即0.0001个ICP。
* --memo <memo>：指定此交易的数字备忘录。

###### to

指定要将 ICP 代币转移到的帐户地址。

##### 示例

1. dfx ledger --network ic transfer dfx ledger --network ic transfer dd81336dbfef5c5870e84b48405c7b229c07ad999fdcacb85b9b9850bd60766f --memo 12345 --icp 1

将部分 ICP 余额发送给另一个账户。

#### top-up

使用 dfx ledger 充值命令用将ICP 代币铸造为cycle，并充值到指定canister中。

##### 语法

dfx ledger top-up [FLAGS] [OPTIONS] <canister>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### OPTIONS

* --amount <amount>：指定要将多少ICP铸造为cycle，这些cycle将存入canister。金额可以指定为最多八位小数的数字。
* --e8s <e8s>：指定ICP 令牌的小数单位。可以单独使用此选项或与 --e8s 结合使用。--e8s 5000000表示0.05000000个ICP。
* --icp <icp>：指定ICP 令牌的整数单位。可以单独使用此选项或与 --e8s 结合使用。
* --fee <fee>：指定交易费用。 默认值为 10000 e8s。即0.0001个ICP。
* --max-fee <max-fee>：指定最高交易费用。 默认值为 10000 e8s。即0.0001个ICP。

###### canister

指定要充值的canister标识符。

##### 示例

1. dfx ledger --network ic top-up --icp 1 5a46r-jqaaa-aaaaa-qaadq-cai

将价值 1 ICP 的cycle充值部署在互联网计算机上的canister中。

#### notify

使用 dfx ledger notify 命令通知ledger关于发送到cycles minting canister的交易信息。

仅当 dfx ledger create-canister 或 dfx ledger top-up 成功向ledger发送消息，并且在某个区块高度记录了交易，但由于某种原因后续通知失败时，才应使用此命令。

##### 语法

dfx ledger notify [FLAGS] [OPTIONS] <block-height> <destination-principal>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### OPTIONS

* --max-fee <max-fee>：指定最高交易费用。 默认值为 10000 e8s。即0.0001个ICP。

###### block-height

指定发送交易记录的块高度。

###### destination-principal

指定目标的主体，可以是canister ID或用户主体的文本表示。如果发送交易用于 create-canister 命令，请指定controller主体。 如果发送交易用于top-up，请指定canister ID。

##### 示例

1. dfx ledger --network ic notify 75948 tsqwz-udeik-5migd-ehrev-pvoqv-szx2g-akh5s-fkyqc-zy6q7-snav6-uqe

向ledger发送通知消息以响应记录在块高度 75948 的交易。

# dfx wallet

使用dfx wallet 命令来管理当前身份关联的的cycle钱包。也可以将cycle发送到其他账户的cycle wallet canister。

## 语法

dfx wallet [FLAGS] [OPTIONS] <SUBCOMMAND>

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --network <network>：指定要连接的计算网络。默认使用本地网络。“IC”表示Internet Computer主网。

## 子命令

#### add-controller

使用 dfx wallet add-controller 向当前身份关联的cycle钱包添加controller。

被分配了 Controller 角色的身份拥有最多的特权，可以对所选身份的cycle钱包执行以下操作：

* 重命名cycle钱包。
* 将条目添加到地址簿。
* 添加和删除controller。
* 授权和取消托管人的授权。

controller也是custodian(托管人)，可以执行与该角色相关的以下操作：

* 访问钱包信息。
* 发送cycle。
* 转发call调用。
* 创建canister。

##### 语法

dfx wallet add-controller [FLAGS] <controller>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### controller

指定要添加到当前身份关联的cycle钱包的controller的主体身份标识。

##### 示例

1. dfx wallet –network ic add-controller hpff-grjfd-tg7cj-hfeuj-olrjd-vbego-lpcax-ou5ld-oh7kr-kl9kt-yae

向IC主网上当前身份关联的cycle钱包添加一个新的controller。

#### remove-controller

使用 dfx wallet remove-controller 命令删除当前身份关联的cycle钱包的controller。

##### 语法

dfx wallet add-controller [FLAGS] <controller>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### controller

指定当前身份关联的cycle钱包中的要删除的controller的主体身份标识。

##### 示例

1. dfx wallet –network ic remove-controller dheus-mqf6t-xafkj-d3tuo-gh4ng-7t2kn-7ikxy-vvwad-dfpgu-em25m-2ae

将IC主网上当前身份关联的cycle钱包中的controller移除。

#### controllers

使用 dfx wallet controllers 命令列出当前身份关联的cycle钱包的controllers。

##### 语法

dfx wallet controllers [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx wallet --network ic controllers

列出当前身份关联的cycle钱包的controllers

#### authorize

使用 dfx wallet authorize 命令为当前身份关联的cycle钱包授权一个托管人。

被分配了托管人角色的身份可以对该cycle钱包执行以下操作：

* 访问钱包信息。
* 发送cycle。
* 转发call调用。
* 创建canister。

##### 语法

dfx wallet authorize [FLAGS] <custodian>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### custodian

指定要作为当前身份关联的cycle钱包的托管人的身份主体。

##### 示例

1. dfx wallet --network ic authorize dheus-mqf6t-xafkj-d3tuo-gh4ng-7t2kn-7ikxy-vvwad-dfpgu-em25m-2ae

为当前身份关联的cycle钱包授权一个托管人。

#### deauthorize

使用 dfx wallet deauthorize 命令从当前身份关联的cycle钱包中删除保管人。

注意：如果托管人也是controller，这也将取消其controller的角色。

##### 语法

dfx wallet deauthorize [FLAGS] <custodian>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### custodian

要删除的当前身份关联的cycle钱包的托管人。

##### 示例

1. dfx wallet --network ic deauthorize dheus-mqf6t-xafkj-d3tuo-gh4ng-7t2kn-7ikxy-vvwad-dfpgu-em25m-2ae

从当前身份关联的cycle钱包中删除保管人。

#### custodians

使用 dfx wallet custodians 命令列出当前身份绑定的cycle钱包的托管人。Cycle钱包的controller也为保管人。

##### 语法

dfx wallet custodians [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx wallet --network ic custodians

列出在IC主网上当前身份绑定的cycle钱包的托管人。

#### addresses

使用 dfx wallet addresses命令显示钱包的地址簿。地址条目包含身份主体标识和角色（联系人、托管人或控制器），并且可能包含与地址关联的名称和种类（unknown、user或canister）

##### 语法

dfx wallet addresses [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx wallet --network icaddresses

显示在IC主网上当前身份绑定的cycle钱包的地址簿。

#### balance

使用 dfx wallet balance 命令显示当前身份绑定的cycle 钱包的余额。

##### 语法

dfx wallet balance [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx wallet --network ic balance

显示在IC主网上当前身份绑定的cycle 钱包的余额。

#### set-name

使用 dfx wallet set-name 命令为当前身份绑定的cycle钱包分配一个名称。

##### 语法

dfx wallet set-name [FLAGS] <name>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### name

待设置的cycle钱包的名称。

##### 示例

1. dfx wallet --network ic set-name Terrances\_wallet

为IC主网上当前身份绑定的cycle钱包分配一个名称。

#### name

如果已使用 dfx wallet set-name 命令设置，则使用 dfx wallet name 命令显示当前身份绑定的cycle钱包的名称。

##### 语法

dfx wallet name [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx wallet --network ic name

显示在IC主网上当前身份绑定的cycle钱包的名称。

#### send

使用 dfx wallet send命令将当前身份绑定的cycle钱包的cycle发送至另外一个cycle钱包。

##### 语法

dfx wallet send [FLAGS] <destination> <amount>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### destination

指定目标cycle钱包的wallet canister ID。

###### amount

指定要发送的cycle数量。

##### 示例

1. dfx wallet --network ic send r7inp-6aaaa-aaaaa-aaabq-cai 2000000000

在IC主网上，当前身份绑定的cycle wallet向r7inp-6aaaa-aaaaa-aaabq-cai发送2000000000个cycle。

#### upgrade

使用 dfx wallet upgrade 命令将当前身份绑定的cycle钱包的 Wasm 模块升级为当前DFX 捆绑的 Wasm。

##### 语法

dfx wallet upgrade [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx wallet --network ic upgrade

将IC主网上当前身份绑定的cycle钱包的 Wasm 模块升级为当前DFX 捆绑的 Wasm。

# dfx canister

使用dfx canister 命令来管理canister与互联网计算机区块链的交互。在大多数情况下，在编译程序后使用 dfx canister 命令来管理容器生命周期并执行关键任务，例如调用程序函数。

## 语法

dfx canister [FLAGS] [OPTIONS] <SUBCOMMAND>

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --network <network>：指定要连接的计算网络。默认使用本地网络。“IC”表示Internet Computer主网。
* --wallet <wallet>：使用指定的cycle wallet canister作为消息的发送者。如果未指定，则默认为当前身份绑定的cycle wallet。
* --no-wallet：以用户身份作为消息的发送者。

## 子命令

#### create

使用 dfx canister create命令在 Internet 计算机上创建新的canister。

注意：必须连接到 Internet 计算机网络或在本地运行网络才能运行此命令。此外，必须在项目目录中才能运行该命令。

注意：dfx canister create创建canister需要经由cycle wallet canister，因此当前身份需要绑定cycle wallet或使用—wallet指定。

##### 语法

dfx canister create [FLAGS] [OPTIONS] <canister-name> <controller>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果在dfx.json中定义了多个canister，则允许一次创建多个。注意，必须指定 --all 或单个canister-name。

###### OPTIONS

* -c, --compute-allocation <compute-allocation>：指定canister的计算分配。在[0..100] 范围内。
* --memory-allocation <memory-allocation>：指定canister总共允许使用多少内存。在[0..256 TB] 范围内。
* -with-cycles <with-cycles>：指定要存入新创建的canister的初始cycle数量。数量要包含canister创建的费用。

###### canister-name

指定要创建的canister的名称，该名称需要在dfx.json中定义。

###### controller

指定要设置为新canister的控制器的身份主体标识符。controller决定了谁可以操作此canister。

##### 示例

1. dfx canister --network ic create --with-cycles 8000000000000 --all

在IC主网上为项目创建所有的canister，并且使用初始的8000000000000 cycle。

#### delete

使用 dfx canister delete 命令从本地 Internet 计算机网络或远程 Internet 计算机网络中删除已停止的canister。

注意：必须连接到 Internet 计算机网络或在本地运行网络才能运行此命令。此外，必须在项目目录中才能运行该命令。

##### 语法

dfx canister delete [FLAGS] [OPTIONS] [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：删除 dfx.json 文件中配置的所有canister。 请注意，必须指定 --all 或单个canister名称。

###### canister

指定要删除的canister的名称, 该名称需要在dfx.json中定义。请注意，必须指定canister名称或 --all 选项。

##### 示例

1. dfx canister --network=ic delete --all

删除该项目在 ic Internet 计算机网络上部署的所有canister。

#### install

使用 dfx canister install 命令将编译后的代码安装到在本地运行的 Internet 计算机网络或您可以访问的子网络上运行的canister。

##### 语法

dfx canister install [FLAGS] [OPTIONS] [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果项目 dfx.json中包含多个canister，则允许一次安装多个canister。请注意，必须指定 --all 或单个容器名称。
* --async-call：直接返回响应ID，而不通过轮询副本等待调用完成。

###### OPTIONS

* --argument <argument>：指定要在安装期间传递给canister的参数。
* --argument-type <argument-type>：当调用使用参数时，用来指定参数的数据格数，可选值：[idl, raw]。默认情况下，使用 Candid (idl) 语法为数据值指定参数，如果要将原始字节传递给canister，则可以使用 raw 作为参数类型。
* -m, --mode <mode>：指定安装类型。您可以将canister安装模式设置为以下三个值之一：
  + install：初始安装，在空的canister进行安装。
  + reinstall：重新安装，不会保留原来canister中的持久化数据。
  + upgrade：升级安装，会保留原来canister中的持久化数据。

###### canister

指定要安装的canister的名称, 该名称需要在dfx.json中定义。请注意，必须指定canister名称或 --all 选项。

##### 示例

1. dfx canister --network ic install --all

在IC主网上部署项目中所有使用dfx build编译后的 WebAssembly 到canister中。

1. dfx canister --network ic install hello\_world --async-call

在IC主网上部署hello\_world的WebAssembly 到canister中，并且不等待。

1. dfx canister --network ic install --m reinstall --all

在IC主网上重新安装项目中所有使用dfx build编译后的 WebAssembly 到canister中。

#### uninstall-code

在Internet 计算机网络上删除指定canister中的代码和状态。

##### 语法

dfx canister uninstall-code [FLAGS] [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果项目 dfx.json中包含多个canister，则允许一次卸载多个canister。请注意，必须指定 --all 或单个容器名称。

###### canister

指定要卸载的canister的名称, 该名称需要在dfx.json中定义。请注意，必须指定canister名称或 --all 选项。

##### 示例

1. dfx canister --network ic uninstall-code --all

在IC主网上卸载项目中所有canister。

#### start

使用 dfx canister start 命令重新启动在本地 Internet 计算机网络或远程 Internet 计算机网络上已停止的canister。

在大多数情况下，在停止容器后运行此命令以正确终止任何挂起的请求，这是升级容器的先决条件。

注意，只能从项目目录结构中运行此命令。

##### 语法

dfx canister start [FLAGS] [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果项目 dfx.json中包含多个canister，则允许一次启动多个canister。请注意，必须指定 --all 或单个容器名称。

###### canister

指定要启动的canister的名称, 该名称需要在dfx.json中定义。请注意，必须指定canister名称或 --all 选项。

##### 示例

1. dfx canister --network=ic start --all

在IC主网上启动项目中所有canister。

#### stop

使用 dfx canister stop 命令停止在本地 Internet 计算机网络或远程 Internet 计算机网络上运行的canister。

在大多数情况下，运行此命令以正确终止任何挂起的请求，这是升级容器的先决条件。

注意，只能从项目目录结构中运行此命令。

##### 语法

dfx canister stop [FLAGS] [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果项目 dfx.json中包含多个canister，则允许一次停止多个canister。请注意，必须指定 --all 或单个容器名称。

###### canister

指定要停止的canister的名称, 该名称需要在dfx.json中定义。请注意，必须指定canister名称或 --all 选项。

##### 示例

1. dfx canister --network=ic stop --all

在IC主网上停止项目中所有canister。

#### deposit-cycles

将指定数量的cycle存到指定的canister中。

注意：dfx canister deposit-cycles需要经由cycle wallet canister，因此当前身份需要绑定cycle wallet或使用—wallet指定。

##### 语法

dfx canister deposit-cycles [FLAGS] <cycles> [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果项目 dfx.json中包含多个canister，则允许一次向多个canister存款。请注意，必须指定 --all 或单个容器名称。

###### cycles

指定需要发送的cycle个数，这个cycle将从指定的wallet中删除。

###### canister

指定要存储的目标canister的名称或ID。请注意，必须指定canister或 --all 选项。

##### 示例

1. dfx canister --network=ic deposit-cycles --all 100000

在IC主网上向所有项目中的canister发送100000cycle。

#### id

使用 dfx canister id 命令输出特定canister名称的canister标识符。

注意，只能从项目目录结构中运行此命令。

##### 语法

dfx canister id [FLAGS] <canister>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### canister

指定canister的名称, 该名称需要在dfx.json中定义。

##### 示例

1. dfx canister --network ic id hello\_world

显示hello\_world canister在IC主网上的canister ID。

#### status

使用 dfx canister status 命令检查在本地 Internet 计算机网络或远程 Internet 计算机网络上的canister当前是否正在运行、正在停止或当前停止。

注意，只能从项目目录结构中运行此命令。

##### 语法

dfx canister status [FLAGS] [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果项目 dfx.json中包含多个canister，则允许一次显示多个canister状态。请注意，必须指定 --all 或单个容器名称。

###### canister

指定目标canister的名称或ID。请注意，必须指定canister或 --all 选项。

##### 示例

1. dfx canister --network=ic status --all

显示在IC主网上的项目中所有canister的状态。

#### info

通过认证的方式获取指定canister的 WASM 模块的哈希值及其controllers。

##### 语法

dfx canister info [FLAGS] <canister>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### canister

指定canister的名称或ID。

##### 示例

1. dfx canister --network ic info hello\_world

获取在IC主网上的hello\_world canister的 WASM 模块的哈希值及其controllers。

#### update-settings

更新canister的一项或多项设置，例如：controller、计算分配、内存分配。

注意，只能从项目目录结构中运行此命令。

##### 语法

dfx canister update-settings [FLAGS] [OPTIONS] [canister]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --all：如果项目 dfx.json中包含多个canister，则允许一次更新多个canister设置。请注意，必须指定 --all 或单个容器名称。

###### OPTIONS

* -c, --compute-allocation <compute-allocation>：指定canister的计算分配。在[0..100] 范围内。
* --memory-allocation <memory-allocation>：指定canister总共允许使用多少内存。在[0..256 TB] 范围内。
* --controller <controller>：指定canister的新controller。

###### canister

指定canister的名称, 该名称需要在dfx.json中定义。

##### 示例

1. dfx canister --no-wallet --network ic update-settings -c 50 --all

更新在IC主网上的项目的所有canister的计算分配为50。

#### sign

在使用 dfx canister send 命令发送消息之前使用 dfx canister sign 命令，这相当于使用单个 dfx canister call 命令。使用单独的调用可以增加事务的安全性。

##### 语法

dfx canister sign [FLAGS] [OPTIONS] <canister-name> <method-name> [argument]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --query：指定向canister发送查询请求。这种类型的请求不会更改canister状态，因此任意canister能够很快返回结果。
* --update：指定向canister发送更新请求。这种类型的请求会更改canister状态，因此需要经过2/3节点共识。

###### OPTIONS

* --expire-after <expire-after>：指定消息的有效发送时间，以秒为单位，默认为 300s（5 分钟）。
* --file <file>：指定输出文件名。默认为 message.json。
* --random <random>：指定生成随机参数的配置。
* --type <type>：用来指定参数的数据格数，可选值：[idl, raw]。默认情况下，使用 Candid (idl) 语法为数据值指定参数，如果要将原始字节传递给canister，则可以使用 raw 作为参数类型。

###### canister-name

指定要调用的canister的名称。canister名称是必需参数，并且必须在dfx.json中指定。

###### method-name

指定要在canister上调用的方法名称。 canister method是必需的参数。

###### argument

指定传递给canister的参数。

##### 示例

1. dfx canister --no-wallet --network ic sign --expire-after=1h rno2w-sqaaa-aaaaa-aaacq-cai create\_neurons ‘(“PUBLIC\_KEY”)’

使用与您使用隐私增强邮件 (PEM) 文件创建的身份关联的主体创建签名的 message.json 文件。

#### send

在使用 dfx canister send 命令发送消息之前使用 dfx canister sign 命令，这相当于使用单个 dfx canister call 命令。使用单独的调用可以增加事务的安全性。

##### 语法

dfx canister send [FLAGS] <file-name>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --status：在消息中发送已签名的request-status调用。

###### file-name

指定已经签名的消息文件。

##### 示例

1. dfx canister --network ic send message.json

发送消息。

#### call

使用 dfx canister call 命令在已部署的canister上调用指定的方法。

##### 语法

dfx canister call [FLAGS] [OPTIONS] <canister-name> <method-name> [argument]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。
* --query：指定向canister发送查询请求。这种类型的请求不会更改canister状态，因此任意canister能够很快返回结果。
* --update：指定向canister发送更新请求。这种类型的请求会更改canister状态，因此需要经过2/3节点共识。
* --async：直接返回响应ID，而不通过轮询副本等待调用完成。

###### OPTIONS

* --output <output>：指定显示方法的返回结果时使用的输出格式。 有效值为idl 和raw。
* --with-cycles <with-cycles>：指定要在调用中发送的cycle数量。将从指定钱包中扣除。
* --random <random>：指定生成随机参数的配置。
* --type <type>：用来指定参数的数据格数，可选值：[idl, raw]。默认情况下，使用 Candid (idl) 语法为数据值指定参数，如果要将原始字节传递给canister，则可以使用 raw 作为参数类型。

###### method-name

指定要在canister上调用的方法名称。canister method是必需的参数。

###### argument

指定传递给canister的参数。

##### 示例

1. dfx canister --network ic call hello --type idl greet '("Lisa")'

使用idl参数执行函数调用。

1. dfx canister --network ic call hello --type raw greet '4449444c00017103e29883'

使用raw参数执行函数调用。

#### request-status

使用 dfx canister request-status 命令请求对canister的指定调用的状态。

此命令要求您指定在调用canister上的方法后收到的请求标识符（调用时使用了--async选项）。请求标识符是一个以 0x 开头的十六进制字符串。

##### 语法

dfx canister request-status [FLAGS] [OPTIONS] <request-id> <canister>

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### OPTIONS

* --output <output>：指定调用返回结果的显示格式，可选的值：idl、raw、pp。

###### request-id

指定dfx canister call或 dfx canister install 命令使用--async选项返回的十六进制字符串。 此标识符是一个以 0x 开头的十六进制字符串。

###### canister

指定发出请求的canister的名称或 ID。

如果向管理canister发出请求，请指定它正在更新/查询的canister的 ID。

如果调用是由钱包代理的，即`dfx canister --wallet=<ID> call --async` 标志，请指定钱包的canister ID。

##### 示例

1. dfx canister --network ic request-status 0x58d08e785445dcab4ff090463b9e8b12565a67bf436251d13e308b32b5058608 hello\_world

发起调用状态查询。

# dfx build

使用 dfx build 命令将您的程序编译成可以部署在 Internet 计算机上的 WebAssembly 模块。

dfx build 命令使用dfx.json 配置文件的 canisters 部分下配置的信息查找要编译的源代码。

注意：只能从项目目录结构中运行此命令。

## 语法

dfx build [FLAGS] [OPTIONS] [canister-name]

## 选项

### FLAGS

* --check：使用临时的、硬编码的、本地定义的canister容器来测试程序在没有连接到 Internet 计算机网络的情况下编译。
* --all：如果在dfx.json中定义了多个canister，则允许一次编译多个。注意，必须指定 --all 或单个canister-name。
* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --network <network>：指定要连接的计算网络。默认使用本地网络。“IC”表示Internet Computer主网。

### canister-name

指定要编译的canister的名称，该名称需要在dfx.json中定义。

## 示例

1. dfx build --network ic hello\_world

在IC主网上编译hello\_world项目。

# dfx deploy

使用 dfx deploy 命令在本地或指定网络上注册、构建和部署应用程序。默认情况下，会部署dfx.json 配置文件中定义的所有canister。

此命令以下命令的简化：

|  |
| --- |
| dfx canister create --all  dfx build  dfx canister install --all |

注意：只能从项目目录结构中运行此命令。

## 语法

dfx deploy [FLAGS] [OPTIONS] [canister-name]

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --network <network>：指定要连接的计算网络。默认使用本地网络。“IC”表示Internet Computer主网。
* --wallet <wallet>：使用指定的cycle wallet canister作为消息的发送者。如果未指定，则默认为当前身份绑定的cycle wallet。
* --no-wallet：以用户身份作为消息的发送者。
* -with-cycles <with-cycles>：指定要存入新创建的canister的初始cycle数量。数量要包含canister创建的费用。
* --argument <argument>：指定要在安装期间传递给canister的参数。
* --argument-type <argument-type>：当调用使用参数时，用来指定参数的数据格数，可选值：[idl, raw]。默认情况下，使用 Candid (idl) 语法为数据值指定参数，如果要将原始字节传递给canister，则可以使用 raw 作为参数类型。

### canister-name

指定要注册、编译、部署的canister的名称，该名称需要在dfx.json中定义。

## 示例

1. dfx deploy --network ic hello

在IC主网上注册、编译、部署hello canister。

# dfx upgrade

使用 dfx upgrade 命令升级本地计算机上运行的 DFINITY Canister SDK 组件。

此命令根据 manifest.json 文件中指定的最新公开可用版本检查您当前安装的 DFINITY Canister SDK 的版本。 如果在本地检测到旧版本的 DFINITY Canister SDK，dfx upgrade 命令会自动从 CDN 获取最新版本。

## 语法

dfx upgrade [FLAGS] [OPTIONS]

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### OPTIONS

* --current-version <version>：指定要标识为当前版本的版本。

## 示例

1. dfx upgrade

进行版本升级。

# dfx cache

使用dfx cache 命令来管理 dfx 版本缓存。

## 语法

dfx cache [FLAGS] <SUBCOMMAND>

## 选项

### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

### 子命令

#### install

使用 dfx cache install 命令使用当前在 dfx 缓存中找到的版本安装 dfx。

##### 语法

dfx cache install [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx cache install

使用当前在 dfx 缓存中找到的版本安装 dfx。

#### delete

使用 dfx cache delete 命令从本地计算机的版本缓存中删除指定版本的 dfx。

##### 语法

dfx cache delete [FLAGS] [OPTIONS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。

###### OPTIONS

* --version <version>：指定要删除的版本。

##### 示例

1. dfx cache delete --version 0.6.2

删除指定版本的dfx版本缓存。

#### list

使用 dfx cache list 命令列出你当前在项目中安装和使用的 dfx 版本。

如果安装了多个 dfx 版本，缓存列表将显示一个星号 (\*) 以指示当前活动的版本。

##### 语法

dfx cache list [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx cache list

打印缓存的dfx版本信息。

#### show

使用 dfx cache show 命令显示当前dfx 版本使用的缓存的完整路径。

##### 语法

dfx cache show [FLAGS]

##### 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

##### 示例

1. dfx cache show

显示当前dfx 版本使用的缓存的完整路径。

# dfx config

使用 dfx config 命令查看或配置当前项目的配置文件中的设置。

注意，只能从项目目录结构中运行此命令。

## 语法

dfx config [FLAGS] [OPTIONS] <config-path> [value]

## 选项

###### FLAGS

* -h, --help：打印帮助信息。
* -V, --version：打印版本信息。

###### OPTIONS

* --format <format>：指定配置文件输出的格式。默认情况下，文件使用 JSON 格式显示。有效值为 json 和 text。

###### config-path

指定要设置或读取的配置选项的名称。必须使用其句点划定的路径来指定配置文件选项。

如果没有指定特定配置选项的路径，该命令将显示完整的配置文件。

###### value

指定要更改的选项的新值。如果未指定值，该命令将从配置文件中返回选项的当前值。

## 示例

1. dfx config canisters.hello.main "src/hello\_world/hello-main.mo"

修改hello canister的编译入口。