Cardano-cli:~$ Study sheets (العربية)

الجزء 1: المفاتيح والعناوين والتفويض

تم تصميم هذا البرنامج التعليمي ليتم استخدامه مع ملخص للإصدار *Cardano-cli (قابل للطباعة) cheat sheet V8.0.0*

تمت الترجمة بواسطة BRNMXR [HAWAK]
شكرا جزيلا لعملك الجيد

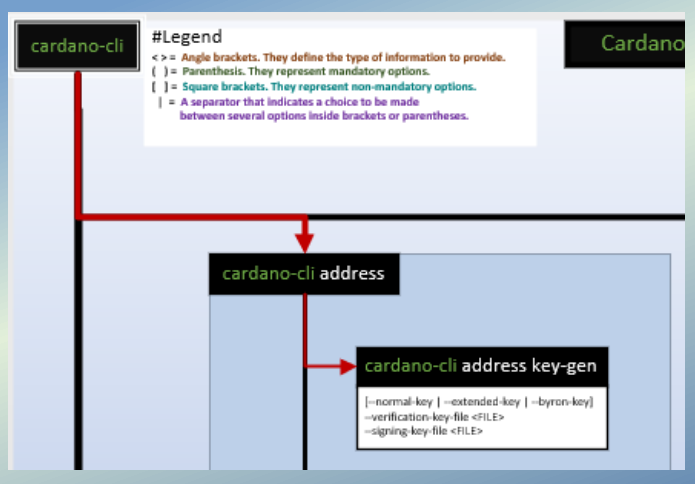
الفجوة الهوائية = العقدة الهوائية غير متصل بالإنترنت

العقدة النشطة = العقدة النشطة (متصل بالإنترنت)

هذه الوثيقة تستهدف شرح تفصيلي لكيفية تفسير أوامر *cardano-cli* وخياراتها المتعلقة من أجل أن تكون قادرًا على تجميعها بنفسك إذا لزم الأمر. للقيام بذلك، يجب أن يكون لديك جهاز كمبيوتر وتقوم بتثبيت عقدة *Cardano Blockchain* و واجهة سطر الأوامر *Cardano (cardano-cli)*. ستبدأ بالأوامر البسيطة وستزداد تعقيدا تدريجياً مع تقدم البرنامج التعليمي.

الفجوة الهوائية — التمرين الأول: إنشاء مفاتيح الدفع والحصة

1 أولاً, حدد الأمر الذي ستستخدمه لمفاتيح الدفع الخاصة بك



2 لديك 5 خيارات بالمجمل

[الخيارات الثلاث الأولى موجودة بين قوسين مربعين مع فاصلين، مما يشير إلى أن الاختيار بين هذه الخيارات الثلاث ليس إلزاميًا، حيث سيتم استخدام مفتاح عادي تلقائياً إذا لم يتم تحديد أي شيء. في هذا المثال، لن تستخدمها.]

cardano-cli address key-gen

~~[--normal-key | --extended-key | --byron-key]~~
--verification-key-file <FILE>
--signing-key-file <FILE>

3 الخيارات التالية يجب استخدامها

الأقواس الزاوية تشير إلى نوع المعلومات <ملف>. يجب عليك توفير الاسم الذي ستعطيه لملفي المفتاح الخاص والعام الخاص بك.

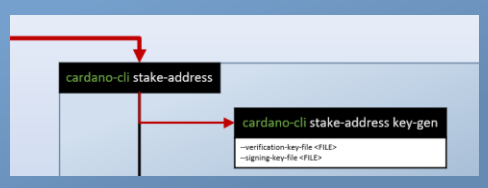
cardano-cli address key-gen

~~[--normal-key | --extended-key | --byron-key]~~
--verification-key-file **payment.vkey**
--signing-key-file **payment.skey**

4 هذه هي النتيجة النهائية لهذا الأمر البسيط على الطرفية الخاصة بك.

```
user@computer:~$ cardano-cli address key-gen \
> --verification-key-file payment.vkey \
> --signing-key-file payment.skey
```

5 الآن بعد أن أصبحت مفاتيح الدفع الخاصة بك جاهزة، عليك إنشاء مفاتيح الكيان. هذه المرحلة أسهل بكثير لأنه يوجد نوع واحد فقط من مفتاحالحصة.



cardano-cli stake-address key-gen

--verification-key-file <FILE>
--signing-key-file <FILE>

6 كما في النقطة 3 ،يجب عليك تحديد نوع المعلومات. <ملف>.

cardano-cli stake-address key-gen

--verification-key-file **stake.vkey**
--signing-key-file **stake.skey**

7 وكنتيجة نهائية على الواجهة...

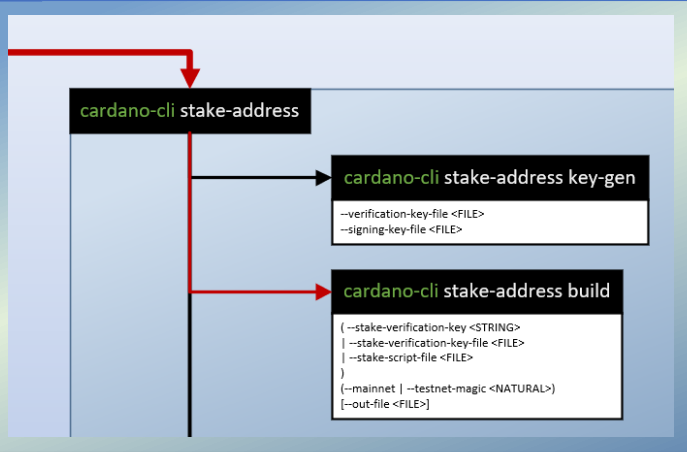
```
user@computer:~$ cardano-cli stake-address key-gen \
> --verification-key-file stake.vkey \
> --signing-key-file stake.skey
```

⚠ تنبيه. يوصى بإنشاء واستخدام مفاتيح الدفع والرهان الخاصة بك لتوقيع المعاملات في بيئة "الفجوة الهوائية" لأسباب أمنية.
https://developers.cardano.org/docs/get-started/air-gap

عندما يتم إنشاء زوجي المفاتيح الخاصة بك، ستكون قادرًا على إنشاء عنوان للحصة يمكنك من خلاله التحقق من مقدار المكافآت الخاصة بك وسحبها عند استخدامها في عملية تحويل مع مفتاح الحصة الخاص بك*(stake.skey)*.

الفجوة الهوائية — التمرين الثاني: إنشاء عنوان للحصة

1 حدد الأمر الذي ستستخدمه لإنشاء عنوان الحصة الخاصة بك



2 لديك 6 خيارات بالمجمل

الخيارات الثلاثة الأولى محصورة بين قوسين وتحتوي على فاصلين، مما يشير إلى أنه يجب اتخاذ اختيار بين هذه الخيارات الثلاثة.

cardano-cli stake-address build

(--stake-verification-key <STRING>
| --stake-verification-key-file <FILE>
| --stake-script-file <FILE>
)
(--mainnet | --testnet-magic <NATURAL>)
[--out-file <FILE>]

3 استخدم الخيار stake-verification-key-file

الأقواس الزاويّة تشير إلى نوع المعلومات <ملف>. في هذه المرة، يجب أن تقدم المسار الذي يشير إلى stake.vkey الخاص بك

cardano-cli stake-address build

~~(--stake-verification-key <STRING>~~
| --stake-verification-key-file **stake.vkey**
~~| --stake-script-file <FILE>~~
)
(--mainnet | --testnet-magic <NATURAL>)
[--out-file <FILE>]

4 لديك الآن خياران بين الأقواس الدائرية.

سيتعين عليك تحديد الشبكة المستخدمة وإذا كانت تستخدم شبكة الاختبار (تيستنت) ، فيجب عليك ذكر رقم الشبكة السحري. في هذه الحالة، سنستخدم الشبكة الرئيسية (ماينت).

cardano-cli stake-address build

--stake-verification-key-file stake.vkey
--mainnet | ~~--testnet-magic <NATURAL>~~)
[--out-file <FILE>]

5 والخيار الأخير --out-file <FILE>

هذا الخيار ليس إلزاميًا لأنه محاط بقوسين مربعين. ومع ذلك ، إذا لم تستخدم هذا الخيار ، فسيتم عرض الإخراج (عنوان الحصة) من الأمر على واجهة الطرفية بدلاً من حفظه في ملف. و بما انك ستحتاج إلى استخدام هذا العنوان للحصة في وقت لاحق ، فلنسميه ونحفظه في ملف.

cardano-cli stake-address build

--stake-verification-key-file stake.vkey
--mainnet
--out-file **stake.addr**

6 هكذا يظهر الأمر على واجهة الطرفية الخاصة بك.

```
user@computer:~$ cardano-cli stake-address build \
> --stake-verification-key-file stake.vkey \
> --mainnet \
> --out-file stake.addr
```

7 ها هو ما يجب أن يكون لديك حتى الآن.

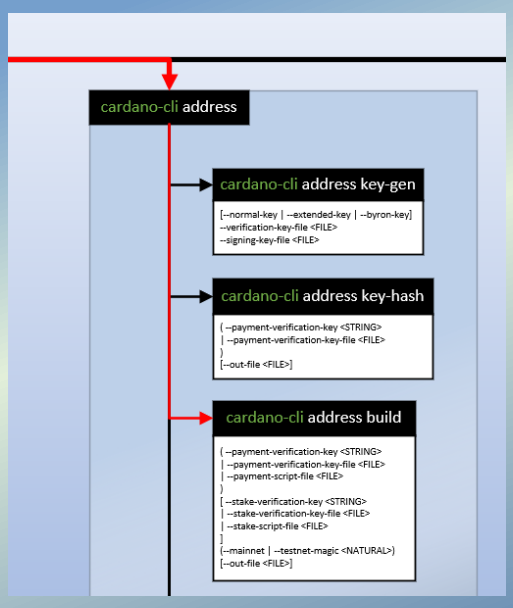
```
user@computer:~$ ls
payment.vkey    payment.skey    stake.vkey
stake.skey      stake.addr
```

"بمجرد إنشاء زوجي المفاتيح الخاصتين بك وعنوان الحصة الخاص بك، ستكون قادرًا على إنشاء عنوان عن طريق دمج مفتاح الدفع الخاص بك مع مفتاح الحصة الخاص بك بحيث يتم تضمين الرصيد في العنوان المُولَّد في بروتوكول الحصة مع مكافآتك."

الفجوة الهوائية

التمرين الثالث: إنشاء دفعة باستخدام ملف عنوان

1 حدد الأمر الذي ستستخدمه لعملية الدفع الخاصة بك باستخدام ملف عنوان الحصة.



2 "لديك 9 خيارات بالمجمل"

يمكنك إنشاء عنوان الدفع باستخدام مفتاح الدفع الخاص بك فقط دون ربطه بمفتاح الحصة الخاص بك، وهذا يفسر سبب إلزامية المجموعة الأولى من الخيارات وليس المجموعة الثانية.

cardano-cli address build

```
( --payment-verification-key <STRING>
| --payment-verification-key-file <FILE>
| --payment-script-file <FILE>
)
[ --stake-verification-key <STRING>
| --stake-verification-key-file <FILE>
| --stake-script-file <FILE>
]
(--mainnet | --testnet-magic <NATURAL>)
[--out-file <FILE>]
```

3 لنستخدم كل من مفتاح التحقق من الدفع ومفتاح التحقق من الحصة مرة أخرى، وفقًا لعلامات الاقتباس الزاوية<FILE> كإيلاء بجيد ، تحديد مسارات ملفات payment.vkey و stake.vkey

cardano-cli address build

~~(--payment-verification-key <STRING>~~
| **--payment-verification-key-file payment.vkey**
~~| --payment-script-file <FILE>~~
~~)~~
~~[--stake-verification-key <STRING>~~
| **--stake-verification-key-file stake.vkey**
~~| --stake-script-file <FILE>~~
]
(--mainnet | --testnet-magic <NATURAL>)
[--out-file <FILE>]

4 الآن ستحتاج إلى تحديد الشبكة المستخدمة واسم الملف لعنوانك.

cardano-cli address build

```
--payment-verification-key-file payment.vkey
--stake-verification-key-file stake.vkey
(--mainnet | --testnet-magic <NATURAL>)
--out-file paymentwithstake.addr
```

5 هذه هي النتيجة النهائية

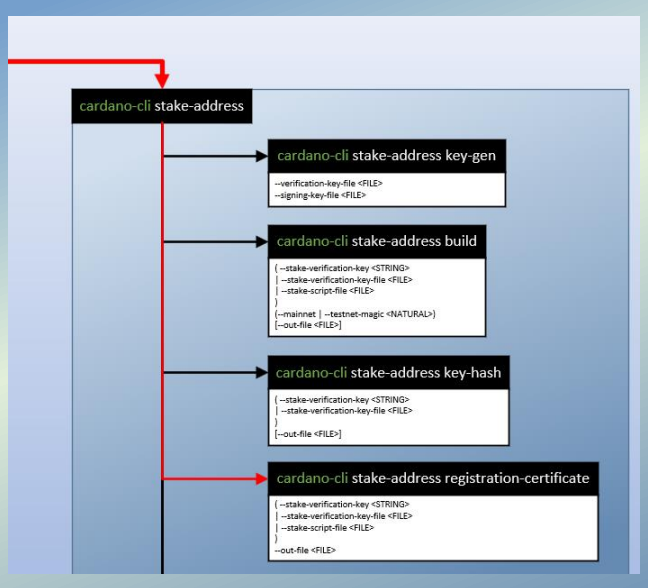
```
user@computer:~$ cardano-cli address build \
> --payment-verification-key-file payment.vkey \
> --stake-verification-key-file stake.vkey \
> --mainnet \
> --out-file paymentwithstake.addr
```

يمكنك نسخ محتوى *"paymentwithstake.addr"* في محرر نصوص ولصقه في عملية تحويل في محفظة كاردانو التي تستخدمها عادةً وإرسال بعض الـ*ADA* إليها. *(يجب أن تكون 10 ADA كافية للبدء)*

الفجوة الهوائية

التمرين الرابع: إنشاء شهادة المساهمة

1 تحديد القسم الذي ستستخدمه لشهادة الحصة الخاصة بك.



2 لديك 4 خيارات

"للمشاركة في البروتوكول ووضعADA الخاص بك في الاستثمار، يجب عليك ربط مفتاح التحقق الخاص بالاستثمار بشهادة ستقوم بإرسالها إلى الblock chain في التمارين المقبلة. الأمر لإنشاء الشهادة الخاصة بك بسيط بما فيه الكفاية. يجب عليك فقط توفير إحدى هذه الخيارات الثلاث الإلزامية وتحديد اسم الملف الذي سيلعب دور الشهادة."

cardano-cli stake-address registration-certificate

```
( --stake-verification-key <STRING>
| --stake-verification-key-file <FILE>
| --stake-script-file <FILE>
)
--out-file <FILE>
```

cardano-cli stake-address registration-certificate

~~(--stake-verification-key <STRING>~~
| **--stake-verification-key-file stake.vrf**
~~| --stake-script-file <FILE>~~
)
--out-file stake.cert

3 هذه هي النتيجة النهائية لكيفية ظهور الأمر على واجهة الطرفية الخاصة بك.

```
user@computer:~$ cardano-cli stake-address registration-certificate \
> --stake-verification-key-file stake.vkey \
> --out-file stake.cert
```

4 هنا ما يجب أن يتوفر عندك حتى الآن.

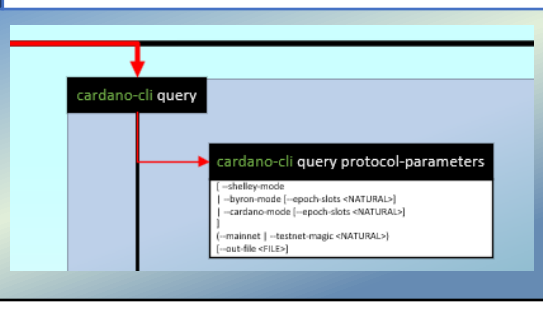
```
user@computer:~$ ls
payment.vkey        payment.skey        stake.vkey        stake.skey
stake.addr          paymentwithstake.addr   stake.cert
```

الآن ستحصل على مؤشرات البروتوكول ونهاية الـ*Blockchain* لـ*(tip)* لتتمكن من البدء في بناء أول عملية تحويل لك.

العقدة النشطة

التمرين الخامس: الحصول على إعدادات البروتوكول

1 قبل كل شيء، ستحتاج إلى إعدادات البروتوكول لحساب الرسوم المتعلقة بعملية التحويل الخاصة بك.



2 لديك 6 خيارات رئيسية و 2 خيارات فرعية بالمجموع.

cardano-cli query protocol-parameters

```
[ --shelley-mode
| --byron-mode [--epoch-slots <NATURAL>]
| --cardano-mode [--epoch-slots <NATURAL>]
]
(--mainnet | --testnet-magic <NATURAL>)
[--out-file <FILE>]
```

3 انتقل إلى خيارات "الوضع". اذكر الشبكة المطلوبة واسم الملف الذي يجب إنشاؤه.

cardano-cli query protocol-parameters

~~[--shelley-mode~~
~~| --byron-mode [--epoch-slots <NATURAL>]~~
~~| --cardano-mode [--epoch-slots <NATURAL>]~~
~~]~~
(--mainnet ~~| --testnet-magic <NATURAL>)~~
--out-file protocol.json

4 هذه هي النتيجة النهائية لكيفية ظهور هذا الأمر على واجهة الطرفية الخاصة بك.

```
user@computer:~$ cardano-cli query protocol-parameters \
> --mainnet \
> --out-file protocol.json
```

5 الآن قم بالمتابعة ولنرى محتوى ذلك الملف:

```
user@computer:~$ cat protocol.json
```

في ملف protocol.json ،ستبحث عن الوديعة التي يجب إجراؤها على block chain لتسجيل عنوانالحصة الخاص بك والمشاركة في بروتوكول الحصص. يمكن استرداد هذه الوديعة في أي وقت إذا قمت بإلغاء تسجيل عنوانك.

6 احتفظ بتسجيل المبلغ المودع، حيث ستحتاجه لاحقًا. يتم التعبير عن المبلغ بوحدة Lovelace. 1 ADA = 1,000,000 لوفلايس

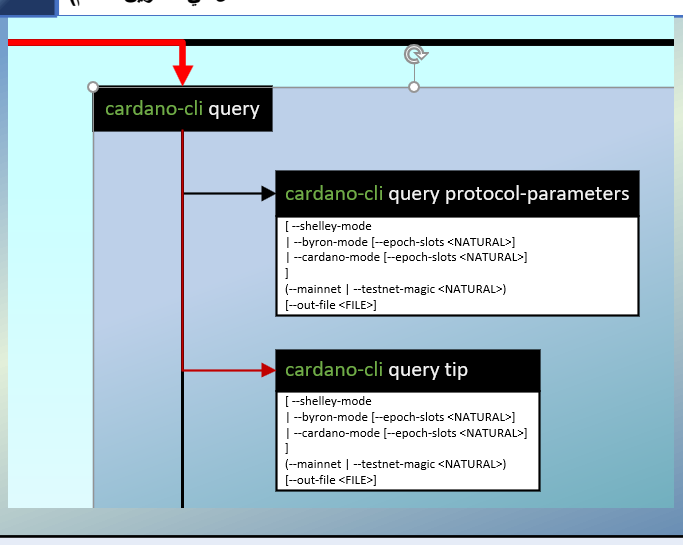

```
"poolRetireMaxEpoch": 18,
"protocolVersion": {
    "major": 8,
    "minor": 0
},
"stakeAddressDeposit": 2000000,
"stakePoolDeposit": 500000000,
"stakePoolTargetNum": 500,
"treasuryCut": 0.2,
"txFeeFixed": 155381,
"txFeePerByte": 44,
"utxoCostPerByte": 4310,
"utxoCostPerWord": null
```


7 الآن عليك أن تأخذ ملف protocol.jsonالخاص بك وتنقله إلى بيئة "الفجوة الجوية" الخاصة بك لتتمكن من حساب الرسوم عند بناء معاملاتك.

⚠ "بتنفيذCIP-1694واقتراب عصر فولتير ، سيصبح من الممكن لحائزي ADA في المجتمع ، بمساعدة اللجنةالدستورية والدرييس ، تعديل معلمات البروتوكول من خلال نظام تصويت منظم. لهذا السبب ، من المهم التأكد من أن لديك أحدث التعديلات لهذه البروتوكولات في بيئتة الـ"Air Gap" الخاصة بك ، حيث يمكن أن يكون لذلك تأثير مباشر على الإعدادات المختلفة المحيطة بمعاملاتك."

العقدة النشطة — التمرين السادس: الحصول على رقم الفتحة

1 في التمرين القادم ستحتاج إلى معرفة نهايةblock chain الحالية للعقدة(tip) لحساب مدة الحياة المتبقية(TTL) لدينا. (سيتم توفير الوصف المفصل في التمرين القادم)



2 مثلما في التمرين السابق، ستكون لديك 6 خيارات و خيارين فرعيين.

الآن بما أنك بدأت في فهم المبدأ بشكل كامل، يمكنك تخطي بعض الخطوات. ليس من الضروري إنشاء ملف، كل ما تحتاجه هو تحديد موقعك في الجدول.

cardano-cli query tip

~~[--shelley-mode~~
~~| --byron-mode [--epoch-slots <NATURAL>]~~
~~| --cardano-mode [--epoch-slots <NATURAL>]~~
~~]~~
(--mainnet | ~~--testnet-magic <NATURAL>~~)
~~[--out-file <FILE>]~~

3 هذه هي النتيجة النهائية على جهازك.

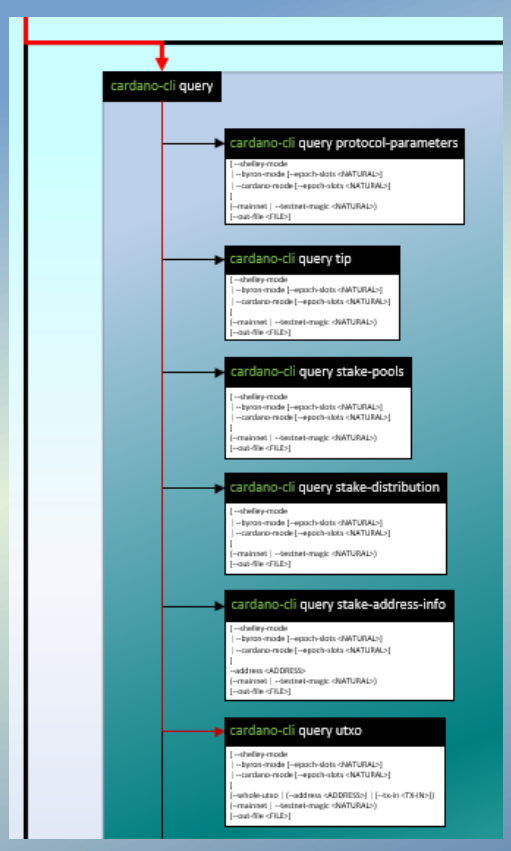
```
user@computer:~$ cardano-cli query tip \
> --mainnet
```

4 لاحظ وجود موقعك في الجدول.


```
{
  "block": 8749125,
  "epoch": 410,
  "era": "Babbage",
  "hash": "503e4af96abc18e1d4b5de08e0d35cb508e364...",
  "slot": 92027764,
  "syncProgress": "100.00"
}
```


العقدة النشطة — التمرين السابع: استعلامUTXO(النقاط الغير صريحة المنقولة)

1 ستقوم الآن باستعلامUTXO لعنوان paymentwithstake.addr(إذا قمت بإرسال ADA إليه



2 هذا الأمر له 9 خيارات وخيارين فرعيين

ستحتاج إلى استهلاكUTXO واحدة على الأقل كإدخال للمعاملة الخاصة بك. يمكن أن تحتوي المعاملة على عدة مداخل ومخرجات مختلفة ، ولكن في هذه الحالة يجب أن يكون لديكUTXOواحد فقط مرتبط بعنوان الدفع الخاص بك (paymentwithstake.addr)لأنك قمت بإيداع 10 ADAفقط في هذا العنوان. لذا سنستخدم فقط ما هو مطلوب. بإختصار ، سنستخدم paymentwithstake.addrالخاص بك والشبكة المستخدمة ، ومن ثم سنقوم بإنشاء ملف لنقل قائمة UTXOهذه إلى بيئة الفجوة الهوائية الخاصة بك "air gap".

cardano-cli query utxo

~~[--shelley-mode~~
~~| --byron-mode [--epoch-slots <NATURAL>]~~
~~| --cardano-mode [--epoch-slots <NATURAL>]~~
~~]~~
~~(--whole-utxo~~ | (--address <ADDRESS>) | ~~(--tx-in <TX-IN>)~~)
(--mainnet | ~~--testnet-magic <NATURAL>~~)
[--out-file <FILE>]

3 هذا ما يجب أن يظهر على واجهة الطرفية الخاصة بك.

cardano-cli query utxo

~~[--shelley-mode~~
~~| --byron-mode [--epoch-slots <NATURAL>]~~
~~| --cardano-mode [--epoch-slots <NATURAL>]~~
~~]~~
~~(--whole-utxo~~ | (--address paymentwithstake.addr) | ~~(--tx-in <TX-IN>)~~)
(--mainnet | ~~--testnet-magic <NATURAL>~~)
[--out-file UTXO.addrs]

```
user@computer:~$ cardano-cli query utxo \
> --address paymentwithstake.addr
> --mainnet
> --out-file utxo.addrs
```

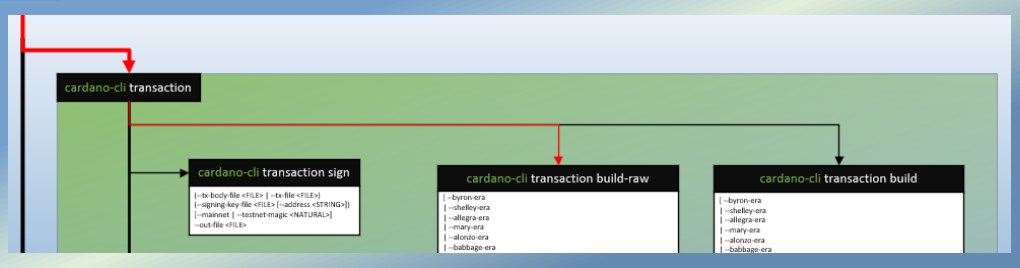
4 يجب أن يظهر محتوى ملفutxo.addr على النحو التالي. يتوافق UTXOلإيداعك بقيمة 10 ADA مع10,000,000لوفليس. يمكنك الآن أخذ هذا الملف وإدخاله في بيئة "الفجوة الهوائية" الخاصة بك. ستحتاج إليه قريبًا.

```
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
1234a4d18e9dkhb34234kjbvdec3ad81e299c1a523443453561e61ce9bf8608e8c802df3b7f8c     0        10000000 lovelace + TxOutDatumNone
```

"الآن حان الوقت لإنشاء أول عملية تحويل لك، سيتم استخدامها لإرسال شهادةالحصة الخاصة بك. قبل أن تبدأ، قد يبدو لك ما ستراه مرهقًا، ولكن عند اتباع الخطوات خطوة بخطوة، يجب أن تكون قادرًا على فهم السبب وكيفية تقليص الخيارات التالية إلى مجموعة من 6 خيارات لعملية التحويل. لأسباب أمان، في هذا البرنامج التعليمي، ستستخدم طرق تنطوي على الأمر *'cardano-cli transaction build-raw'* بدلاً من الأمر *'cardano-cli transaction build'* ، حيث يمكن إنشاؤه في بيئة غير متصلة بالإنترنت."

فجوة جوية **التمرين الثامن:** إنشاء مسودة لأول عملية تحويل خاصة بك

1 ابحث عن أمر "cardano-cli transaction build-raw"



2 "لنبدأ تدريجياً من الأعلى إلى الأسفل."

```
cardano-cli transaction build-raw
[ --byron-era
| --shelley-era
| --allegra-era
| --mary-era
| --alonzo-era
| --babbage-era
]
```

3 [الخيارات الأولى الخمسة] اختيارية وموضوعة بين [أقواس مربعة]. إذا لم يتم تحديد أي شيء، سيتم استخدام عصر ماري بشكل افتراضي.

```
cardano-cli transaction build-raw
[ --byron-era
| --shelley-era
| --allegra-era
| --mary-era
| --alonzo-era
| --babbage-era
]
[--script-valid | --script-invalid]
```

4 (لا تحتوي عمليتك على نص، لذا يمكنك تجاوز الخيارين التاليين.)

```
cardano-cli transaction build-raw
[ --byron-era
| --shelley-era
| --allegra-era
| --mary-era
| --alonzo-era
| --babbage-era
]
[--script-valid | --script-invalid]
```

5 لشرح الخيارات التالية و20 خيارًا فرعيًا لها، يلزم توفير شرح مفصل.

داخل قوس اختياري، يمكن أن تكون هناك خيارات فرعية محددة بواسطة الأعمدة. ولهذا السبب،يوجد مفهوم الأولوية وترتيب معين يجب احترامه عند بناء عملية نقل تراكمية. في هذه الحالة، نعلم أن هناك 3 أعمدة مميزة تحدد الترتيب الذي يجب إدراج الخيارات به إذا كنا نرغب في إنتاج هيكل (الجسم) للعملية بشكل صحيح.

```
1 [--script-valid | --script-invalid]
  (--tx-in <TX-IN>
2   [--spending-tx-in-reference <TX-IN>
      --spending-plutus-script-v2
3     ( --spending-reference-tx-in-datum-cbor-file
      | --spending-reference-tx-in-datum-file
```

6 خذ الوقت اللازم لتحليل ترتيب أولويات الخيار"tx-in" والأقواس المقابلة بعناية

```
[--script-valid | --script-invalid]
(--tx-in <TX-IN>
 [ --spending-tx-in-reference <TX-IN>
  --spending-plutus-script-v2
  ( --spending-reference-tx-in-datum-cbor-file <CBOR FILE>
  | --spending-reference-tx-in-datum-file <JSON FILE>
  | --spending-reference-tx-in-datum-value <JSON VALUE>
  | --spending-reference-tx-in-inline-datum-present
  )
  ( --spending-reference-tx-in-redeemer-cbor-file <CBOR FILE>
  | --spending-reference-tx-in-redeemer-file <JSON FILE>
  | --spending-reference-tx-in-redeemer-value <JSON VALUE>
  )
  --spending-reference-tx-in-execution-units (<INT, INT>)
 | --simple-script-tx-in-reference <TX-IN>
 | --tx-in-script-file <FILE>
  [
   ( --tx-in-datum-cbor-file <CBOR FILE>
   | --tx-in-datum-file <JSON FILE>
   | --tx-in-datum-value <JSON VALUE>
   | --tx-in-inline-datum-present
   )
   ( --tx-in-redeemer-cbor-file <CBOR FILE>
   | --tx-in-redeemer-file <JSON FILE>
   | --tx-in-redeemer-value <JSON VALUE>
   )
   --tx-in-execution-units (<INT, INT>)]
 ])
[--read-only-tx-in-reference <TX-IN>]
[--tx-in-collateral <TX-IN>]
```

7 "tx-in--"مطلوب، ولكن ليس خياراته الفرعية.

"الخيارات التالية ليست إلزامية وتُستخدم في نصوص بلوتوس. لذلك ليس من الضروري استخدامها في عملية النقل البسيطة لدينا."

```
--tx-in <TX-IN>
 [ --spending-tx-in-reference <TX-IN>
  --spending-plutus-script-v2
  ( --spending-reference-tx-in-datum-cbor-file <CBOR FILE>
  | --spending-reference-tx-in-datum-file <JSON FILE>
  | --spending-reference-tx-in-datum-value <JSON VALUE>
  | --spending-reference-tx-in-inline-datum-present
  )
  ( --spending-reference-tx-in-redeemer-cbor-file <CBOR FILE>
  | --spending-reference-tx-in-redeemer-file <JSON FILE>
  | --spending-reference-tx-in-redeemer-value <JSON VALUE>
  )
  --spending-reference-tx-in-execution-units (<INT, INT>)
 | --simple-script-tx-in-reference <TX-IN>
 | --tx-in-script-file <FILE>
  [
   ( --tx-in-datum-cbor-file <CBOR FILE>
   | --tx-in-datum-file <JSON FILE>
   | --tx-in-datum-value <JSON VALUE>
   | --tx-in-inline-datum-present
   )
   ( --tx-in-redeemer-cbor-file <CBOR FILE>
   | --tx-in-redeemer-file <JSON FILE>
   | --tx-in-redeemer-value <JSON VALUE>
   )
   --tx-in-execution-units (<INT, INT>)]
 ])
```

8 لما يلي:

لن تحتاج إلى "إدخال مرجعي للقراءة فقط" أو أي شيء آخر يتعلق بالضمانات، حيث أنها عملية بسيطة لن تتضمن نص بلوتوس.

```
[--read-only-tx-in-reference <TX-IN>]
[--tx-in-collateral <TX-IN>]
[--tx-out-return-collateral <ADDRESS VALUE>]
[--tx-total-collateral <INTEGER>]
[--required-signer <FILE> | --required-signer-hash <HASH>]
[--tx-out <ADDRESS VALUE>
  [ --tx-out-datum-hash <HASH>
  | --tx-out-datum-hash-cbor-file <CBOR FILE>
```

9 Circa "--required-signer-hash <HASH>" عن

[لن يكون هذا الخيار مفيدًا في الوقت الحالي لمعاملتك ، والذي سيتم استخدامه لإرسال شهادة حصتك على سلسلة الكتل ، ولكن من المهم ملاحظة أنه سيكون مفيدًا للغاية فيما يتعلق بالتدريب على التصويت على الحكم.]

```
[--read-only-tx-in-reference <TX-IN>]
[--tx-in-collateral <TX-IN>]
[--tx-out-return-collateral <ADDRESS VALUE>]
[--tx-total-collateral <INTEGER>]
[--required-signer <FILE> | --required-signer-hash <HASH>]
[--tx-out <ADDRESS VALUE>
```

10 أخيرًا! خيار تحتاجه "tx-out--"

"سوف تحتاج إلى هذا الخيار لتحديد العنوان الذي سيتم إرسال رصيد UTXO الخاص بك إليه، بعد خصم الرسوم. لذلك دعنا ننسخ هذا الخيار ونضيفه إلى مسودة عملية النقل الخاصة بك ."

```
[--read-only-tx-in-reference <TX-IN>]
[--tx-in-collateral <TX-IN>]
[--tx-out-return-collateral <ADDRESS VALUE>]
[--tx-total-collateral <INTEGER>]
[--required-signer <FILE> | --required-signer-hash <HASH>]
--tx-out <ADDRESS VALUE>
```

11 يمكن تجاوز الخيارات الفرعية لـ "tx-out" المتعلقة بنص بلوتوس. لست بحاجةإليها في الوقت الحالي.

```
--tx-out <ADDRESS VALUE>
[ --tx-out-datum-hash <HASH>
| --tx-out-datum-hash-cbor-file <CBOR FILE>
| --tx-out-datum-hash-file <JSON FILE>
| --tx-out-datum-hash-value <JSON VALUE>
| --tx-out-datum-embed-cbor-file <CBOR FILE>
| --tx-out-datum-embed-file <JSON FILE>
| --tx-out-datum-embed-value <JSON VALUE>
| --tx-out-inline-datum-cbor-file <CBOR FILE>
| --tx-out-inline-datum-file <JSON FILE>
| --tx-out-inline-datum-value <JSON VALUE>
]
[--tx-out-reference-script-file <FILE>]]
[--mint <VALUE>
```

12 أنت تبدأ تدريجيا في فهمها. لا يوجد أصول متعددة، لا يوجد رموز غير قابلة للاستبدال، لا يوجد نص بلوتوس لمعاملاتك.

```
[--mint <VALUE>
 (--mint-script-file <FILE>
  [
   (--mint-redeemer-cbor-file <CBOR FILE>
   |--mint-redeemer-file <JSON FILE>
   |--mint-redeemer-value <JSON VALUE>
   )
   --mint-execution-units (<INT, INT>)]
 |--simple-minting-script-tx-in-reference <TX-IN> --policy-id <HASH>
 |--mint-tx-in-reference <TX-IN>
  --mint-plutus-script-v2
  (--mint-reference-tx-in-redeemer-cbor-file <CBOR FILE>
  |--mint-reference-tx-in-redeemer-file <JSON FILE>
  |--mint-reference-tx-in-redeemer-value <JSON VALUE>
  )
  --mint-reference-tx-in-execution-units (<INT, INT>)
  --policy-id <HASH>
 )]
```

13 ستحتاج إلى استخدام 3 من الخيارات الأربعة التالية

- "يحدد في أي موقع في الجدول ستكون التحويلة صالحة للمعالجة. --invalid-before"
- " بينما"--invalid-herafter" يحدد في أيموقع في الجدول ستصبح التحويلة غير صالحة. (تمامًا مثلتاريخ الانتهاء)

```
[--invalid-before <SLOT>]
[--invalid-hereafter <SLOT>]
[--fee <LOVELACE>]
[--certificate-file <CERTIFICATEFILE>
```

14 احصل على TTLوالرسوم وملف الشهادة.

"لهذا السبب، قمت بتنفيذ الأمر "cardano-cli query tip" في بعض التمارين السابقة. من خلال معرفةالموقع في الجدول للعقدة لعملية 'TTLالمتزامنة لديك، يمكنك تحديد 'الوقت المتبقي للاستخدام' أو ' التحويل الخاصة بك أثناء وجودها في مجموعة الذاكرة المؤقتة. وبالتالي، يمكنك الآن إضافة هذه الخيارات الثلاثة إلى مسودتك وسيتم شرحها لاحقًا."

```
[--invalid-before <SLOT>]
 --invalid-hereafter <SLOT>]
 --fee <LOVELACE>]
 --certificate-file <CERTIFICATEFILE>
```

15 مرة أخرى، لن يتم استخدام أي خيارات متعلقة بشهادات نصوص بلوتوس.

```
--certificate-file <CERTIFICATEFILE>
[--certificate-script-file <FILE>
  [
   (--certificate-redeemer-cbor-file <CBOR FILE>
   |--certificate-redeemer-file <JSON FILE>
   |--certificate-redeemer-value <JSON VALUE>
   )
   --certificate-execution-units (<INT, INT>)]
 |--certificate-tx-in-reference <TX-IN>
  --certificate-plutus-script-v2
  (--certificate-reference-tx-in-redeemer-cbor-file <CBOR FILE>
  |--certificate-reference-tx-in-redeemer-file <JSON FILE>
  |--certificate-reference-tx-in-redeemer-value <JSON VALUE>
  )
  --certificate-reference-tx-in-execution-units (<INT, INT>)
 ]]
```

16 الخيار "withdrawal--"هو إدخال يتيح لك سحب مكافآتك من حساب الحصة الخاصة بك.

```
[--withdrawal <WITHDRAWAL>
 [ --withdrawal-script-file <FILE>
  [
   ( --withdrawal-redeemer-cbor-file <CBOR FILE>
   | --withdrawal-redeemer-file <JSON FILE>
   | --withdrawal-redeemer-value <JSON VALUE>
   )
   --withdrawal-execution-units (<INT, INT>)]
 | --withdrawal-tx-in-reference <TX-IN>
  --withdrawal-plutus-script-v2
  ( --withdrawal-reference-tx-in-redeemer-cbor-file <CBOR FILE>
  | --withdrawal-reference-tx-in-redeemer-file <JSON FILE>
  | --withdrawal-reference-tx-in-redeemer-value <JSON VALUE>
  )
  --withdrawal-reference-tx-in-execution-units (<INT, INT>)
 ]]
[--json-metadata-no-schema | --json-metadata-detailed-schema]
[--auxiliary-script-file <FILE>]
[--metadata-json-file <FILE> | --metadata-cbor-file <FILE>]
[--genesis <FILE> | --protocol-params-file <FILE>]
[--update-proposal-file <FILE>]
--out-file <FILE>
```

17 يمكنك تجاوز الخيار"withdrawal--"وأي خيارات متعلقة بنص بلوتوس في الوقت الحالي.

```
[--withdrawal <WITHDRAWAL>
 [ --withdrawal-script-file <FILE>
  [
   ( --withdrawal-redeemer-cbor-file <CBOR FILE>
   | --withdrawal-redeemer-file <JSON FILE>
   | --withdrawal-redeemer-value <JSON VALUE>
   )
   --withdrawal-execution-units (<INT, INT>)]
 | --withdrawal-tx-in-reference <TX-IN>
  --withdrawal-plutus-script-v2
  ( --withdrawal-reference-tx-in-redeemer-cbor-file <CBOR FILE>
  | --withdrawal-reference-tx-in-redeemer-file <JSON FILE>
  | --withdrawal-reference-tx-in-redeemer-value <JSON VALUE>
  )
  --withdrawal-reference-tx-in-execution-units (<INT, INT>)
 ]]
[--json-metadata-no-schema | --json-metadata-detailed-schema]
[--auxiliary-script-file <FILE>]
[--metadata-json-file <FILE> | --metadata-cbor-file <FILE>]
[--genesis <FILE> | --protocol-params-file <FILE>]
[--update-proposal-file <FILE>]
--out-file <FILE>
```

18 تبقى عدد قليل من الخيارات.

"
في الوقت الحالي ، ليس لديك بيانات وصفية لتقديمها ، ولا ملف نصي مساعد ، ولا حاجة لتحديد ملف البداية أو معلمات البروتوكول. بالإضافة إلى ذلك ، لا ترسل مقترح تحديث لصندوق
ما تبقى لك هو ببساطة إمكانية
Catalyst إعطاء اسم لملف مسودتك للمعاملة
(--out-file <FILE>).
"

```
[--json-metadata-no-schema | --json-metadata-detailed-schema]
[--auxiliary-script-file <FILE>]
[--metadata-json-file <FILE> | --metadata-cbor-file <FILE>]
[--genesis <FILE> | --protocol-params-file <FILE>]
[--update-proposal-file <FILE>]
--out-file <FILE>
```

19 عند تجميع الخيارات التي قمت بنسخها خلال هذا التمرين، ستحصل على شيء مماثل لهذا:

cardano-cli transaction build-raw

```
--tx-in <TX-IN>
--tx-out <ADDRESS VALUE>
[--invalid-hereafter <SLOT>]
[--fee <LOVELACE>]
[--certificate-file <CERTIFICATEFILE>
--out-file <FILE>
```

20 الآن حان وقت إكمال المسودة الخاصة بك:

أضِف إدخال UTXO (tx-in)يقابلا نأوندعو ، (tx-out)وملف الشهادة. في الوقت الحالي، قم بتعيين قيمة0 ل.fee و invalid-hereafter و tx-out

cardano-cli transaction build-raw

```
--tx-in 1234a4d18e9dkhb34234kjbvdec3ad81e299c#0
--tx-out $(cat paymentwithstake.addr)+0
--invalid-hereafter 0
--fee 0
--certificate-file stake.cert
--out-file tx.raw
```

21 إليك كيف سيظهر على محطتك النهائية:

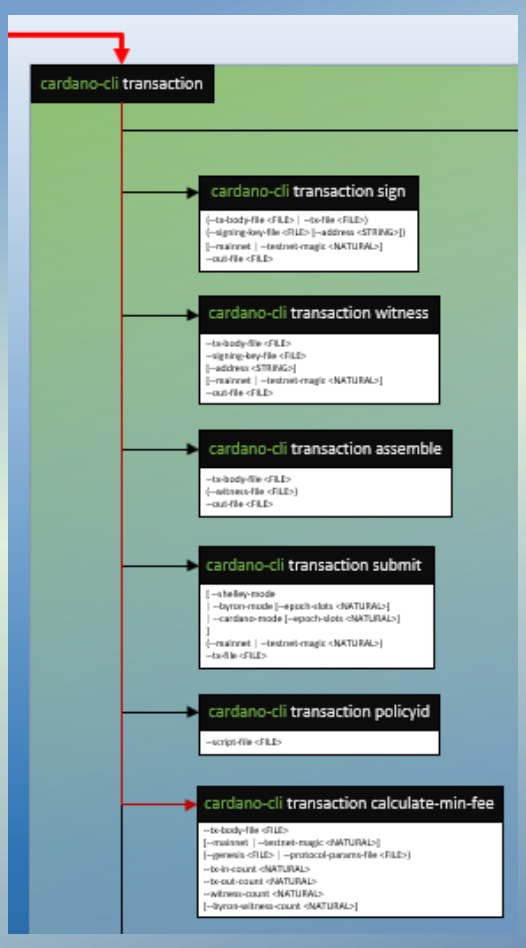
```
user@computer:~$ cardano-cli transaction build-raw \
> --tx-in 1234a4d18e9dkhb34234kjbvd81e299c#0 \
> --tx-out $(cat paymentwithstake.addr)+0 \
> --invalid-hereafter 0 \
> --fee 0 \
> --certificate-file stake.cert \
> --out-file tx.raw
```

تهانينا! لقد وصلت. احفظ الأمر والخيارات من النقطة 21 في محرر نصوص ملف، ستحتاج إليها بعد التمرين التالي. الآن ستحسب الرسوم التي ستكلفك عملية التحويل الخاصة بك. ثم يمكنك خصمها من مبلغ ***UTXO*** الخاص بك ولا تنسى تضمين الإيداع لتسجيل عنوان الحصة.

الفجوة الهوائية

التمرين الثاني: حساب الرسوم (العمولات)

1 ابحث عن الفرع الذي ستستخدمه لحساب الرسوم (المصاريف).



2 لديك 9 خيارات بالمجمل

"tx-inسيقدم لك هذا الأمر مبلغ الرسوم الذي يجب عليك دفعه استنادًا إلى عدد tx-out " وعدد التوقيعات المطلوبة.

cardano-cli transaction calculate-min-fee

```
--tx-body-file <FILE>
[--mainnet | --testnet-magic <NATURAL>]
(--genesis <FILE> | --protocol-params-file <FILE>)
--tx-in-count <NATURAL>
--tx-out-count <NATURAL>
--witness-count <NATURAL>
[--byron-witness-count <NATURAL>]
```

3 من هذه الخيارات لن يتم استخدامها.

- testnet-magic(سنستخدم بالطبع الشبكة الرئيسية في هذا البرنامج التعليمي)
- genesis (سوف تستخدمإعدادات البروتوكول التي حصلت عليها سابقًا)
- byron-witness-count(لماذا لا تستخدم أزواج مفاتيح بايرن)

cardano-cli transaction calculate-min-fee

```
--tx-body-file <FILE>
[--mainnet | --testnet-magic <NATURAL>]
(--genesis <FILE> | --protocol-params-file <FILE>)
--tx-in-count <NATURAL>
--tx-out-count <NATURAL>
--witness-count <NATURAL>
[--byron-witness-count <NATURAL>]
```

4 طفق تارايخلا هذه نم ةثلاث مادختسا متيس ،.

يرجى تحديد عدد عناوين الإدخال والإخراج، بالإضافة إلى عدد المفاتيح التي ستستخدمها لتوقيع عملية التحويل الخاصة بك.

cardano-cli transaction calculate-min-fee

```
--tx-body-file <FILE>
[--mainnet | --testnet-magic <NATURAL>]
(--genesis <FILE> | --protocol-params-file <FILE>)
--tx-in-count 1
--tx-out-count 1
--witness-count 2
[--byron-witness-count <NATURAL>]
```

5 ثم يجب عليك ببساطة تحديد المسار إلى ملفprotocol.jsonالخاص بك ومسودة المعاملة tx.raw

cardano-cli transaction calculate-min-fee

```
--tx-body-file tx.raw
--mainnet
--protocol-params-file protocol.json
--tx-in-count 1
--tx-out-count 1
--witness-count 2
```

6 هذه هي النتيجة في وحدة التحكم الخاصة بك. (قيمة الرسوم قد لا تكون ثابتة دائمًا.)

```
user@computer:~$ cardano-cli transaction calculate-min-fee \
> --tx-body-file tx.raw \
> --mainnet \
> --protocol-params-file protocol.json \
> --tx-in-count 1 \
> --tx-out-count 1 \
> --witness-count 2
```

إذا عمل الأمر كما هو متوقع، ستظهر الرسوم في الجزء السفلي منه.

```
user@computer:~$ cardano-cli transaction calculate-min-fee \
> --tx-body-file tx.raw \
> --mainnet \
> --protocol-params-file protocol.json \
> --tx-in-count 1 \
> --tx-out-count 1 \
> --witness-count 2
178525 Lovelace
```

للتمرين التالي، ستحتاج إلى فتح الملف باستخدام محرر النص الذي قمت بحفظه سابقًا باستخدام الأمر ***"cardano-cli transaction build-raw"*** في التمرين الثامن. ستقوم بتعديل محتواه لإنشاء المعاملة النهائية الخاصة بك.

الفجوة الجوية

التمرين العاشر: بناء المعاملة النهائية

1 هذا هو مسودة عملية النقل الخاصة بك في التمرين الثامن.

ستقوم بتعديلها لإدخال مبلغ الرسوم (التي تعرفها) غلبم بـاسحب موقتسم مڈ ، لوفليس لإرساله إلى عنوانك.

cardano-cli transaction build-raw

```
--tx-in 1234a4d18e9dkhb34234kjbvdec3ad81e299c#0
--tx-out $(cat paymentwithstake.addr)+0
--invalid-hereafter 0
--fee 178525
--certificate-file stake.cert
--out-file tx.raw
```

2 باستخدام الأمر"expr" يمكنك تنفيذ الحساب.

كمية UTXO — عنوان إيداع الحصص

```
user@computer:~$ expr 10000000 - 178525 – 2000000
```

Fee

```
user@computer:~$ expr 10000000 - 178525 – 2000000
7821475
user@computer:~$
```

3 يمكنك إدخال النتيجة في عملية التحويل الخاصة بك.

تذكر أنه لا يجب أن يكون هناك مسافة بين عنوانك وعامل '+' وكمية اللوفليس. وإلا، سيحدث خطأ أثناء تنفيذ الأمر.

cardano-cli transaction build-raw

```
--tx-in 1234a4d18e9dkhb34234kjbvdec3ad81e299c#0
--tx-out $(cat paymentwithstake.addr)+7821475
--invalid-hereafter 0
--fee 178525
--certificate-file stake.cert
--out-file tx.raw
```

4 نحدد الآن "TTL"الخاص بك (الوقت المتبقي للاستخدام)

"لتحديد الموقع في الجدول التي ستجعل العملية غير صالحة، يجب عليك معرفة عددالمواقع التي تمر بها، والتي يمكنك الحصول عليها عن طريق تكرار التمرين رقم 6 أو فحص السجلات الخاصة بك. هنا مثال على ما قد تحصل عليه:"

```
{
  "block": 8749178,
  "epoch": 410,
  "era": "Babbage",
  "hash": "367e4af96abc18e1d4b5de08af535cb508e691...",
  "slot": 92029934,
  "syncProgress": "100.00"
}
```

5 أضف بضع دقائق إليها. (كل موقع = ثانية واحدة)

"لتمكينك من الحصول على وقت كافٍ لتوقيع وإرسال عملية النقل الخاصة بك إلى "العقدة النشطة" الخاصة بك، أضف 15 دقيقة إلى قيمة الخيار.
(92029934 + 900 = 92030834)

cardano-cli transaction build-raw

```
--tx-in 1234a4d18e9dkhb34234kjbvdec3ad81e299c#0
--tx-out $(cat paymentwithstake.addr)+7821475
--invalid-hereafter 92030834
--fee 178525
--certificate-file stake.cert
--out-file tx.raw
```

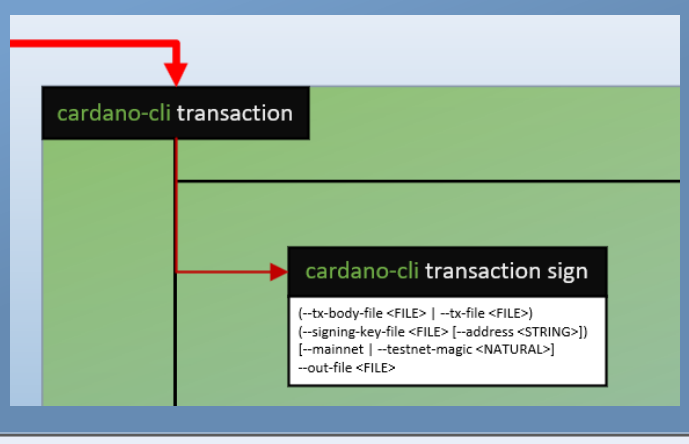
6 هذه هي النتيجة في وحدة التحكم الخاصة بك:

```
user@computer:~$ cardano-cli transaction build-raw \
> --tx-in 1234a4d18e9dkhb34234kjbvdec3ad81e299c#0 \
> --tx-out $(cat paymentwithstake.addr)+7821475 \
> --invalid-hereafter 92030834 \
> --fee 178525 \
> --certificate-file stake.cert \
> --out-file tx.raw
```

الفجوة الجوية

التمرين الحادي عشر: توقيع عملية التحويل

1 الآن أنت مستعد لتوقيع عملية النقل الخاصة بك باستخدام المفاتيح الخاصة بك(payment.skey et stake.skey)



2 لديك مجموعة من 7 خيارات.

لذلك، ستحتاج إلى ذكر مسار "ملف العملية" الخاص بك، و مسارات المفتاحين الخاصين بك، والشبكة المستخدمة، واسم الملف الذي سترسله إلى block chain

cardano-cli transaction sign

```
(--tx-body-file <FILE> | --tx-file <FILE>)
(--signing-key-file <FILE> [--address <STRING>])
[--mainnet | --testnet-magic <NATURAL>]
--out-file <FILE>
```

3 هذه هي النتيجة على جهازك:

```
user@computer:~$ cardano-cli transaction sign\
> --tx-body-file tx.raw \
> --signing-key-file payment.skey \
> --signing-key-file stake.skey \
> --mainnet \
> --out-file tx.signed
```

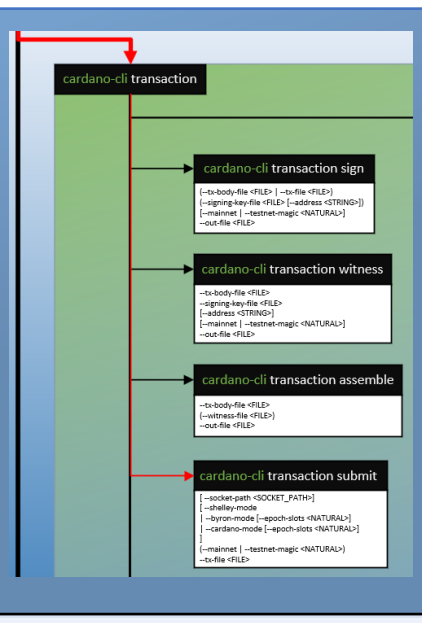
⚠ نلاحظ أنه في العديد من الحالات يمكن استخدام بعض الخيارات أكثر من مرة.

يمكنك الآن نقل الملف *"tx.signed"* إلى "العقدة النشطة" الخاصة بك لإرساله إلى ال *block chain*ةنأب فـلملا اذهل لوصولا نذإ نييعت نم ًلاوأ دكأت نكلو ،"قراءة فقط".

العقدة النشطة

التمرين الثاني عشر: إرسال معاملتك

1 أنت الآن جاهز لإرسال عمليتك المالية!



2 عندك مجموعة من 9 خيارات.

ليس من الضروري استخدام الخيار "--socket-path" إذا كان مسار ملفالسوكت الخاص بعقدتك موجودًا بالفعل في البيئة. ستستخدم فقط ما هو مطلوب. بعبارة أخرى، الشبكة واسم الملف المراد إرساله.

cardano-cli transaction submit

```
[ --socket-path <SOCKET_PATH>]
[ --shelley-mode
| --byron-mode [ --epoch-slots <NATURAL>]
| --cardano-mode [ --epoch-slots <NATURAL>]
]
(--mainnet | --testnet-magic <NATURAL>)
--tx-file <FILE>
```

3 هذا هو الأمر في واجهة الطرفية:

```
user@computer:~$ cardano-cli transaction submit \
> --mainnet \
> --tx-file tx.signed
```

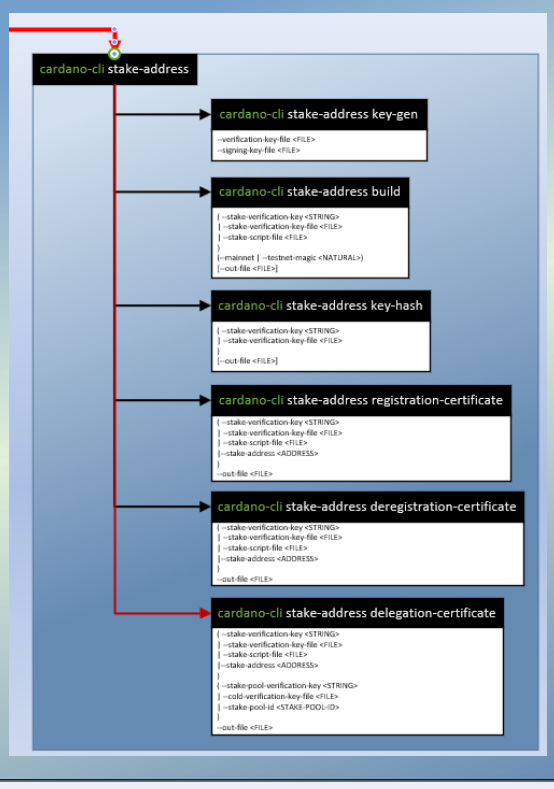
```
user@computer:~$ cardano-cli transaction submit \
> --mainnet \
> --tx-file tx.signed
transaction successfully submitted
```

"مبروك، تم تسجيل عنوان الحصة الخاصة بك على الBlockchainيمكنك الآن إنشاء شهادة وكالة لاختيار مجموعة حصص والمشاركة في بروتوكول *"Proof of Stake"* الخاص بكاردانو. ومع ذلك، قبل المتابعة إلى التمرين التالي، تأكد من حذف ملف *tx.signed* من العقد الخاص بك.
(لن تحتاجه بعد الآن)"

الفجوة الجوية

التمرين الثالث عشر: إنشاء شهادة تفويض

1 قبل كل شيء، حدد الفرع الذي ستستخدمه لشهادتك.



2 "لديك 8 خيارات بالمجمل."

عند إرساله إلى سلسلة الكتل، ستستخدم هذه الشهادة للإشارة إلى المجموعة التي ترغب في ربط عملتك (ADA) بها. هناك مجموعتان من الخيارات الضرورية. ستتم اختيار هذه الخيارات بطرق مختلفة حسب احتياجاتك (على سبيل المثال، باستخدام cold.vrf لمجموعة الرهان الشخصية الخاصة بك). في هذا التمرين، نفترض أنك ترغب في اختيار بين خيارات متنوعة لمجموعة الرهان.

```
cardano-cli stake-address delegation-certificate
( --stake-verification-key <STRING>
| --stake-verification-key-file <FILE>
| --stake-script-file <FILE>
|--stake-address <ADDRESS>
)
( --stake-pool-verification-key <STRING>
| --cold-verification-key-file <FILE>
| --stake-pool-id <STAKE-POOL-ID>
)
--out-file <FILE>
```

3 نستخدم بالتالي --stake-pool-id و --stake-address

- قم بتحديد المسار (PATH) لملف stake.addr.الخاص بك.
- معرّف حوض الحصص الذي ترغب في تفويض محفظتك إليه. (قد يتم ترميزه بتنسيق Bech32أوالسادس عشري)
- اسم ملف الشهادة الخاص بك.

```
cardano-cli stake-address delegation-certificate
( --stake-verification-key <STRING>
| --stake-verification-key-file <FILE>
| --stake-script-file <FILE>
|--stake-address stake.addr
)
( --stake-pool-verification-key <STRING>
| --cold-verification-key-file <FILE>
| --stake-pool-id pool1mt8sdg37f2h3rypyuc77k7vxrjshtvjw04zdjlae9vdzyt9uu34
)
--out-file delegation.cert
```

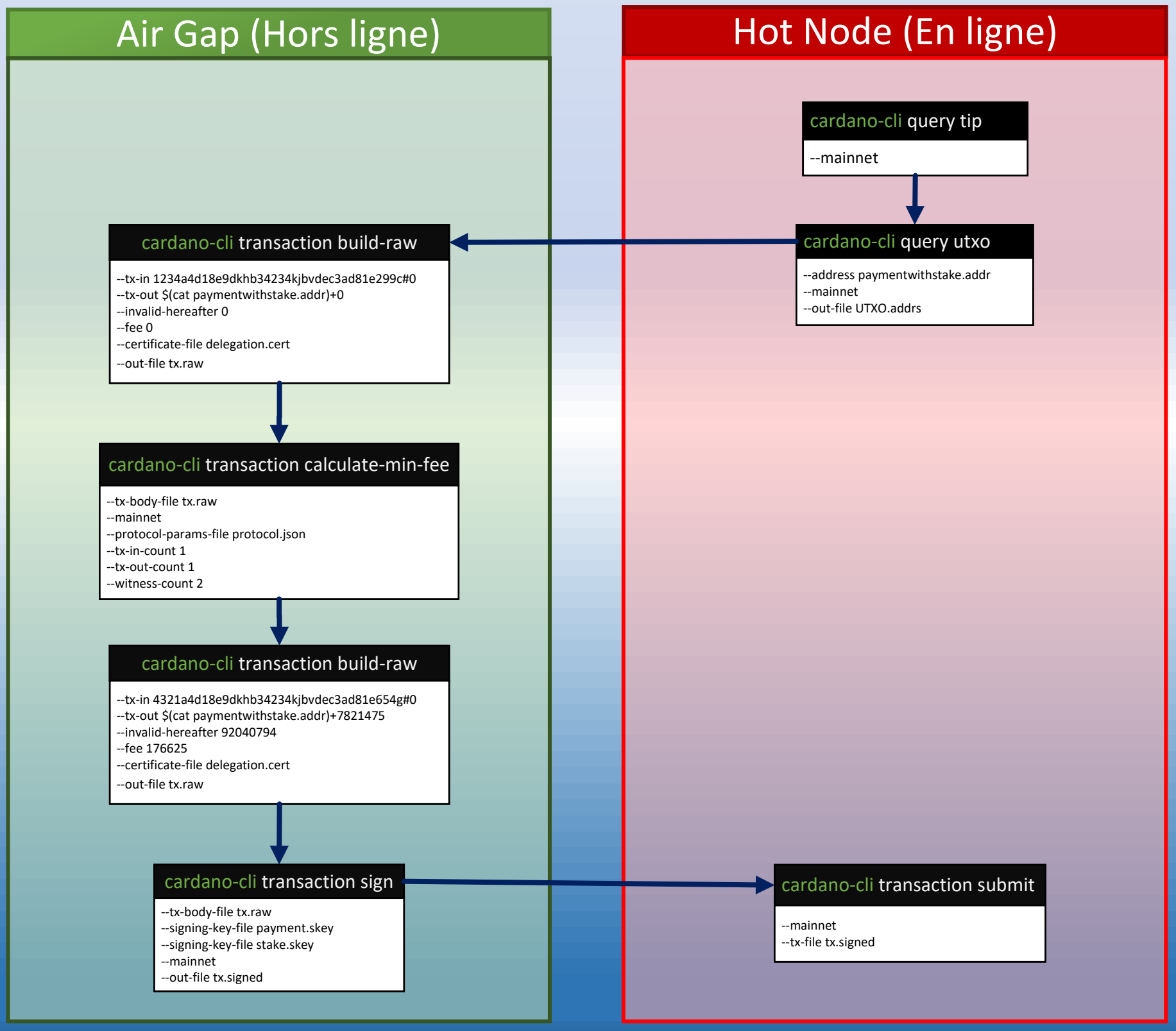
4 هذه هي النتيجة على الطرفية الخاصة بك:

```
user@computer:~$ cardano-cli stake-address delegation-certificate \
> --stake-address stake.addr \
> --stake-pool-id pool1mt8sdg37f2h3rypyuc77k7vxrjshtvjw04zdjlae9vdzyt9uu34 \
> --out-file delegation.cert
```

⚠ Vous pouvez obtenir l'ID du pool de mise sur cexplorer.io ou si vous aimez ce document, faites-le nous savoir, nous pourrions ajouter les commandes : "query stake-pools", "query pool-state" et "query pool-distribution" au 2ème chapitre de ce tutoriel.

يمكنك الآن تكرار التمارين من 6 إلى 12 فلم لادبتسا نم دكأتلا عم ، stakecert بملف Delegation.cert عند بناء عملية التحويل الخاصة بك. ولا تنسى أنه عند حساب الرسوم، يجب عدم احتساب إيداع 2 ADA لعنوان المشاركة (الذي تم إجراؤه بالفعل مسبقًا).

"ملخص العمليات: عملية إرسال شهادة التوكيل"



سنُنهي الجزء الأول من هذا البرنامج التعليمي بعبارة لزميل SPO التي أُعجب بها كثيرًا:

"يجب أن نشجع مشغلي البنية الأساسية المشتركة الجديدة، حتى لو كانت لديهم مهارات منخفضة. سيتعلمون وسوف يتم توزيع Cardano بشكل لا مركزي."

--@StakeWithPride