

From 428ec29e5ac846ba5666f6f61814d21b9a0767b5 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 02:55:46 -0300
Subject: [PATCH 01/30] Create golden.txt to run tests

First:

```
python extract.py card.xls > golden.txt
```

Then run test:

```
diff golden.txt <(python extract.py card.xls)
```

```
diff --git golden.txt golden.txt
```

```
new file mode 100644
```

```
index 0000000..9c7d2ba
```

```
--- /dev/null
```

```
+++ golden.txt
```

```
@@ -0,0 +1,11 @@
```

```
+ BRL COMPRA
```

```
+ 9.01 Amazon Beer Witbier Tapereba
```

```
+ 9.02 Brooklyn Lager
```

```
+ 9.03 DaDo Bier Lager
```

```
+ 9.04 Erdinger Champ
```

```
+ 9.05 Hoegaarden
```

```
+ 9.06 Kirin Ichiban
```

```
+ 9.07 Leffe Blonde
```

```
+ 9.08 Paulistania
```

```
+ 9.09 Rasen Bier Pilsen
```

```
+ 9.1 Therezopolis
```

```
--
```

```
2.3.4
```

From e9d3daf3677c943ae18a32fa3b7f571a5af50931 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:05:24 -0300
Subject: [PATCH 02/30] Extract FIRST_ROW and FIRST_COL

diff --git extract.py extract.py
index d45bf41..ce3b91e 100644

```
--- extract.py
+++ extract.py
@@ -10,2 +10,5 @@ def extract(filename):

+     FIRST_ROW = 2
+     FIRST_COL = 1
+
     headers = []
@@ -13,3 +16,3 @@ def extract(filename):
     while True:
-         value = sheet.cell_value(2, 1 + j)
+         value = sheet.cell_value(FIRST_ROW, FIRST_COL + j)

@@ -27,3 +30,3 @@ def extract(filename):
     while True:
-         value = sheet.cell_value(3 + i, 1 + j)
+         value = sheet.cell_value(FIRST_ROW + 1 + i, FIRST_COL + j)

--
2.3.4
```

From 380cc2a027737242b30a069274668a7f7f910bd7 Mon Sep 17 00:00:00 2001

From: Henrique Bastos <henrique@bastos.net>

Date: Sat, 30 Jul 2016 03:10:14 -0300

Subject: [PATCH 03/30] Replace while with for on headers

diff --git extract.py extract.py

index ce3b91e..f19c1c1 100644

--- extract.py

+++ extract.py

@@ -12,13 +12,9 @@ def extract(filename):

FIRST_COL = 1

+ LAST_ROW = 12

+ LAST_COL = 4

headers = []

- j = 0

- while True:

- value = sheet.cell_value(FIRST_ROW, FIRST_COL + j)

-

- if not value:

- break

-

+ for j in range(FIRST_COL, LAST_COL + 1):

+ value = sheet.cell_value(FIRST_ROW, j)

headers.append(value)

- j += 1

--

2.3.4

From 60ee971b9f7a55733a0f78c18594e1cc2697ecf8 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:11:46 -0300
Subject: [PATCH 05/30] Add viewport.py

diff --git viewport.py viewport.py

new file mode 100644

index 0000000..e69de29

--

2.3.4

From 9531c98bab98a4b5aaac1e2f198e7e3bd06ffff1c Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:15:08 -0300
Subject: [PATCH 06/30] Viewport.__init__

diff --git viewport.py viewport.py
index e69de29..4369cd9 100644

```
--- viewport.py
+++ viewport.py
@@ -0,0 +1,15 @@
+
+
+class Viewport(object):
+    def __init__(self, top, left, bottom, right):
+        self.top = top
+        self.left = left
+        self.bottom = bottom
+        self.right = right
+
+
+if __name__ == '__main__':
+    assert Viewport(2, 1, 12, 4)
+
+    v = Viewport(2, 1, 12, 4)
+    assert v.top == 2 and v.left == 1 and v.bottom == 12 and v.right == 4
--
2.3.4
```

From 58b5c08c7dce9f8d4cd1969d17233702ebf0025b Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:15:16 -0300
Subject: [PATCH 07/30] Viewport.__repr__

diff --git viewport.py viewport.py

index 4369cd9..d80d888 100644

--- viewport.py

+++ viewport.py

@@ -9,2 +9,8 @@ class Viewport(object):

```
+     def __repr__(self):  
+         return '{}({}, {}, {}, {})'.format(  
+             self.__class__.__name__,  
+             self.top, self.left, self.bottom, self.right  
+         )  
+
```

@@ -15 +21,3 @@ if __name__ == '__main__':

assert v.top == 2 and v.left == 1 and v.bottom == 12 and v.right == 4

```
+  
+     assert repr(v) == 'Viewport(2, 1, 12, 4)'
```

--

2.3.4

From 9d4bd67fc5e500b0a49fed812f31fe5b7e5ba554 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:17:53 -0300
Subject: [PATCH 08/30] Viewport.__eq__

```
diff --git viewport.py viewport.py
index d80d888..0e2e482 100644
--- viewport.py
+++ viewport.py
@@ -15,2 +15,13 @@ class Viewport(object):

+    def __str__(self):
+        return '({{}}, {{}}, ({{}}, {{}})'.format(
+            self.top, self.left, self.bottom, self.right
+        )
+
+    def __eq__(self, other):
+        return (self.top == other.top and
+            self.left == other.left and
+            self.bottom == other.bottom and
+            self.right == other.right)
+

@@ -23 +34,4 @@ if __name__ == '__main__':
     assert repr(v) == 'Viewport(2, 1, 12, 4)'
+    assert str(v) == '((2, 1), (12, 4))'
+
+    assert Viewport(2, 1, 12, 4) == Viewport(2, 1, 12, 4)
--
2.3.4
```


From 76dbd1193eb4cdab2f6dcd5fba38944dd5f8599d0 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:30:23 -0300
Subject: [PATCH 09/30] Viewport.rows v1

diff --git viewport.py viewport.py
index 0e2e482..b7989bc 100644

```
--- viewport.py
+++ viewport.py
@@ -26,2 +26,15 @@ class Viewport(object):

+     @property
+     def rows(self):
+         rows_ = []
+
+         for i in range(self.top, self.bottom + 1):
+             row = []
+             for j in range(self.left, self.right + 1):
+                 row.append((i, j))
+
+             rows_.append(tuple(row))
+
+         return tuple(rows_)
+

@@ -37 +50,4 @@ if __name__ == '__main__':
     assert Viewport(2, 1, 12, 4) == Viewport(2, 1, 12, 4)
+
+    assert Viewport(0, 0, 1, 1).rows == ((0, 0), (0, 1)), ((1, 0), (1, 1))
+    assert Viewport(0, 0, 1, 1).rows[0] == ((0, 0), (0, 1))
--
2.3.4
```

From 736796a341be69e91c5276dac8e397198a842d32 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:33:46 -0300
Subject: [PATCH 10/30] Viewport.rows v2 listcomp

diff --git viewport.py viewport.py
index b7989bc..65bfa9e 100644

```
--- viewport.py
+++ viewport.py
@@ -28,12 +28,6 @@ class Viewport(object):
     def rows(self):
-        rows_ = []
+        col_indexes = range(self.left, self.right + 1)
+        row_indexes = range(self.top, self.bottom + 1)

-        for i in range(self.top, self.bottom + 1):
-            row = []
-            for j in range(self.left, self.right + 1):
-                row.append((i, j))
-
-            rows_.append(tuple(row))
-
-        return tuple(rows_)
+        return tuple([tuple([(i, j) for j in col_indexes]) for i in row_indexes])

--
2.3.4
```

From 035284bf539b39f19a01941104985888e89995c0 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:34:13 -0300
Subject: [PATCH 11/30] Viewport.rows v3 genexp

diff --git viewport.py viewport.py

index 65bfa9e..16f187c 100644

--- viewport.py

+++ viewport.py

@@ -31,3 +31,3 @@ class Viewport(object):

- return tuple([tuple([(i, j) for j in col_indexes]) for i in row_indexes])

+ return tuple(tuple((i, j) for j in col_indexes) for i in row_indexes)

--

2.3.4

From 3b9293cb22034b8f658e01e0b6d4306fa7dbc663 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:38:14 -0300
Subject: [PATCH 12/30] Use viewport attrs instead of constants

```
diff --git extract.py extract.py
index 5156993..d28eccc 100644
--- extract.py
+++ extract.py
@@ -4,2 +4,3 @@ from itertools import chain
     import xlrd
+from viewport import Viewport

@@ -10,10 +11,7 @@ def extract(filename):

-    FIRST_ROW = 2
-    FIRST_COL = 1
-    LAST_ROW = 12
-    LAST_COL = 4
+    vp = Viewport(2, 1, 12, 4)

    headers = []
-    for j in range(FIRST_COL, LAST_COL + 1):
-        value = sheet.cell_value(FIRST_ROW, j)
+    for j in range(vp.left, vp.right + 1):
+        value = sheet.cell_value(vp.top, j)
        headers.append(value)
@@ -21,5 +19,5 @@ def extract(filename):
    data = []
-    for i in range(FIRST_ROW + 1, LAST_ROW + 1):
+    for i in range(vp.top + 1, vp.bottom + 1):
        row = []
-        for j in range(FIRST_COL, LAST_COL + 1):
+        for j in range(vp.left, vp.right + 1):
            value = sheet.cell_value(i, j)
--
2.3.4
```

From 9f34f7279e0168321aa69be66d735850c6c54d41 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:39:03 -0300
Subject: [PATCH 13/30] Use Viewport.rows for headers

diff --git extract.py extract.py

index d28eccc..e0a2542 100644

--- extract.py

+++ extract.py

@@ -14,4 +14,4 @@ def extract(filename):

headers = []

- for j in range(vp.left, vp.right + 1):

- value = sheet.cell_value(vp.top, j)

+ for i, j in vp.rows[0]:

+ value = sheet.cell_value(i, j)

headers.append(value)

--

2.3.4

From 939d371f237fea38993d865be505980b3cd8c821 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:40:43 -0300
Subject: [PATCH 14/30] Use Viewport.rows for data

```
diff --git extract.py extract.py
index e0a2542..529f731 100644
--- extract.py
+++ extract.py
@@ -19,5 +19,5 @@ def extract(filename):
     data = []
-    for i in range(vp.top + 1, vp.bottom + 1):
+    for indexes in vp.rows[1:]:
         row = []
-        for j in range(vp.left, vp.right + 1):
+        for i, j in indexes:
             value = sheet.cell_value(i, j)
--
2.3.4
```

From bcd4f27e59559dcf52b6dcefd3e300d65661ccc0 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:42:20 -0300
Subject: [PATCH 15/30] Refactor headers with listcomp

diff --git extract.py extract.py

index 529f731..7440e6e 100644

--- extract.py

+++ extract.py

@@ -13,6 +13,2 @@ def extract(filename):

- headers = []

- for i, j in vp.rows[0]:

- value = sheet.cell_value(i, j)

- headers.append(value)

@@ -26,2 +22,3 @@ def extract(filename):

data.append(row)

+ headers = [sheet.cell_value(i, j) for i, j in vp.rows[0]]

--

2.3.4

From a0083679b1124c05e2cb6dc69413554ea6b7c231 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:43:41 -0300
Subject: [PATCH 16/30] Refactor data with listcomp

diff --git extract.py extract.py

index 7440e6e..668b1b6 100644

--- extract.py

+++ extract.py

@@ -13,13 +13,7 @@ def extract(filename):

```
-  
-     data = []  
-     for indexes in vp.rows[1:]:  
-         row = []  
-         for i, j in indexes:  
-             value = sheet.cell_value(i, j)  
-             row.append(value)  
-  
-     data.append(row)  
-     headers = [sheet.cell_value(i, j) for i, j in vp.rows[0]]  
  
+     data = [[sheet.cell_value(i, j) for i, j in indexes]  
+             for indexes in vp.rows[1:]]  
+  
+     return headers, data  
--  
2.3.4
```


From 08f237c9a0cafdcfae1a1c565ab8fe84a56b6400 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:52:28 -0300
Subject: [PATCH 17/30] RowIterator.__getitem__ for index

```
diff --git viewport.py viewport.py
index 16f187c..29568e3 100644
--- viewport.py
+++ viewport.py
@@ -34,2 +34,14 @@ class Viewport(object):

+class RowIterator:
+    def __init__(self, viewport):
+        self.vp = viewport
+
+    def __getitem__(self, item):
+        col_indexes = range(self.vp.left, self.vp.right + 1)
+        row_index = self.vp.top + item
+
+        return tuple(tuple((row_index, j) for j in col_indexes))
+
+
+
+
+    if __name__ == '__main__':
@@ -47 +59,4 @@ if __name__ == '__main__':
     assert Viewport(0, 0, 1, 1).rows[0] == ((0, 0), (0, 1))
+
+    v = Viewport(0, 0, 3, 1)
+    assert RowIterator(v)[1] == ((1, 0), (1, 1))
--
2.3.4
```

From cdc8a46258be2a181fe27a496919a28f9b2a08e7 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:56:30 -0300
Subject: [PATCH 18/30] RowIterator.__getitem__ for slice

```
diff --git viewport.py viewport.py
index 29568e3..a596ca4 100644
--- viewport.py
+++ viewport.py
@@ -39,2 +39,15 @@ class RowIterator:
     def __getitem__(self, item):
+        if isinstance(item, slice):
+            offset = self.vp.top
+            length = self.vp.bottom + 1 - self.vp.top
+            start, stop, step = item.indices(length)
+
+            slice_ = slice(start + offset, stop + offset, step)
+
+            col_indexes = range(self.vp.left, self.vp.right + 1)
+            row_indexes = range(slice_.start, slice_.stop, slice_.step)
+
+            return tuple((tuple((i, j) for j in col_indexes)
+                               for i in row_indexes))
+        else:
+            col_indexes = range(self.vp.left, self.vp.right + 1)
@@ -62 +75,2 @@ if __name__ == '__main__':
     assert RowIterator(v)[1] == ((1, 0), (1, 1))
+    assert RowIterator(v)[1:-1] == (((1, 0), (1, 1)), ((2, 0), (2, 1)))
--
2.3.4
```

From abc9e589831b96b86ad51249001619f1749f32f7 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 03:59:32 -0300
Subject: [PATCH 19/30] Extract RowIterator._index

```
diff --git viewport.py viewport.py
index a596ca4..9bb4f0a 100644
--- viewport.py
+++ viewport.py
@@ -52,2 +52,5 @@ class RowIterator:
     else:
+        return self._index(item)
+
+    def _index(self, item):
        col_indexes = range(self.vp.left, self.vp.right + 1)
@@ -58,3 +61,2 @@ class RowIterator:

-
-    if __name__ == '__main__':
--
2.3.4
```

From d06f742923c4d7000aafa352741e368838673ef1 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 04:00:07 -0300
Subject: [PATCH 20/30] Extract RowIterator._slice

```
diff --git viewport.py viewport.py
index 9bb4f0a..fcfc087 100644
--- viewport.py
+++ viewport.py
@@ -40,2 +40,13 @@ class RowIterator:
     if isinstance(item, slice):
+         return self._slice(item)
+     else:
+         return self._index(item)
+
+     def _index(self, item):
+         col_indexes = range(self.vp.left, self.vp.right + 1)
+         row_index = self.vp.top + item
+
+         return tuple(tuple((row_index, j) for j in col_indexes))
+
+     def _slice(self, item):
+         offset = self.vp.top
@@ -51,10 +62,2 @@ class RowIterator:
         for i in row_indexes))
-     else:
-         return self._index(item)
-
-     def _index(self, item):
-         col_indexes = range(self.vp.left, self.vp.right + 1)
-         row_index = self.vp.top + item
-
-         return tuple(tuple((row_index, j) for j in col_indexes))
-
--
2.3.4
```

From 02d88d044a50a8a0b026d1d256bfbd9f147a1be6 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 04:00:39 -0300
Subject: [PATCH 21/30] Extract RowIterator.__len__

```
diff --git viewport.py viewport.py
index fcfc087..de26fc7 100644
--- viewport.py
+++ viewport.py
@@ -38,2 +38,5 @@ class RowIterator:

+     def __len__(self):
+         return self.vp.bottom + 1 - self.vp.top
+
         def __getitem__(self, item):
@@ -52,4 +55,3 @@ class RowIterator:
         offset = self.vp.top
-         length = self.vp.bottom + 1 - self.vp.top
-         start, stop, step = item.indices(length)
+         start, stop, step = item.indices(len(self))

--
2.3.4
```

From 3bc344d53aad56f698bee31880fffe4cf8c3ae39 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 04:02:18 -0300
Subject: [PATCH 22/30] Use RowIterator on Viewport.rows

```
diff --git viewport.py viewport.py
index de26fc7..8eac4a6 100644
--- viewport.py
+++ viewport.py
@@ -28,6 +28,3 @@ class Viewport(object):
     def rows(self):
-        col_indexes = range(self.left, self.right + 1)
-        row_indexes = range(self.top, self.bottom + 1)
-
-        return tuple(tuple((i, j) for j in col_indexes) for i in row_indexes)
+        return RowIterator(self)

@@ -77,3 +74,3 @@ if __name__ == '__main__':
-    assert Viewport(0, 0, 1, 1).rows == (((0, 0), (0, 1)), ((1, 0), (1, 1)))
+    assert Viewport(0, 0, 1, 1).rows[:] == (((0, 0), (0, 1)), ((1, 0), (1, 1)))
    assert Viewport(0, 0, 1, 1).rows[0] == ((0, 0), (0, 1))
--
2.3.4
```

From 7be49ac3704fe8d007173536d29ca20cdf597900 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 04:07:51 -0300
Subject: [PATCH 23/30] Refactor extract funcion into a Spreadsheet class

```
diff --git extract.py extract.py
index 668b1b6..99ca658 100644
--- extract.py
+++ extract.py
@@ -7,3 +7,4 @@ from viewport import Viewport

-def extract(filename):
+class Spreadsheet:
+    def __init__(self, filename):
+        book = xlrd.open_workbook(filename, formatting_info=True)
@@ -11,10 +12,16 @@ def extract(filename):

-    vp = Viewport(2, 1, 12, 4)
+    self.sheet = sheet
+    self.vp = Viewport(2, 1, 12, 4)

-    headers = [sheet.cell_value(i, j) for i, j in vp.rows[0]]
+    @property
+    def headers(self):
+        return [self.sheet.cell_value(i, j) for i, j in self.vp.rows[0]]

-    data = [[sheet.cell_value(i, j) for i, j in indexes]
-            for indexes in vp.rows[1:]]
+    @property
+    def data(self):
+        return [[self.sheet.cell_value(i, j) for i, j in indexes]
+                for indexes in self.vp.rows[1:]]

-    return headers, data
+    def __iter__(self):
+        return chain([self.headers], self.data)

@@ -24,5 +31,5 @@ if __name__ == '__main__':

-    headers, data = extract(filename)
+    sheet = Spreadsheet(filename)

-    for row in chain([headers], data):
+    for row in sheet:
+        print('{3: >6} {2}'.format(*row))

--
2.3.4
```

From 37ba20500f34ca5c60e731c5e5bad98358d5fb7c Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 11:50:56 -0300
Subject: [PATCH 24/30] Extract Spreadsheet class to it's own module

diff --git extract.py extract.py
index 99ca658..f0e275d 100644

```
--- extract.py
+++ extract.py
@@ -1,27 +1,4 @@
     import sys
-from itertools import chain

-import xlrd
-from viewport import Viewport
-
-
-class Spreadsheet:
-    def __init__(self, filename):
-        book = xlrd.open_workbook(filename, formatting_info=True)
-        sheet = book.sheet_by_index(0)
-
-        self.sheet = sheet
-        self.vp = Viewport(2, 1, 12, 4)
-
-    @property
-    def headers(self):
-        return [self.sheet.cell_value(i, j) for i, j in self.vp.rows[0]]
-
-    @property
-    def data(self):
-        return [[self.sheet.cell_value(i, j) for i, j in indexes]
-                for indexes in self.vp.rows[1:]]
-
-    def __iter__(self):
-        return chain([self.headers], self.data)
+from spreadsheet import Spreadsheet
```

diff --git spreadsheet.py spreadsheet.py

new file mode 100644

index 0000000..8a56f11

```
--- /dev/null
+++ spreadsheet.py
@@ -0,0 +1,25 @@
+from itertools import chain
+
+import xlrd
+from viewport import Viewport
+
+
+class Spreadsheet:
+    def __init__(self, filename):
+        book = xlrd.open_workbook(filename, formatting_info=True)
+        sheet = book.sheet_by_index(0)
+
+        self.sheet = sheet
+        self.vp = Viewport(2, 1, 12, 4)
+
+    @property
+    def headers(self):
+        return [self.sheet.cell_value(i, j) for i, j in self.vp.rows[0]]
+
```



```
+     @property
+     def data(self):
+         return [[self.sheet.cell_value(i, j) for i, j in indexes]
+                 for indexes in self.vp.rows[1:]]
+
+     def __iter__(self):
+         return chain([self.headers], self.data)
```

```
--
```

2.3.4

From 27053884a5c3ce8c2c703fe9c5086349b7b14c02 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 11:52:19 -0300
Subject: [PATCH 25/30] Apply dependency injection of Viewport on Spreadsheet

```
diff --git extract.py extract.py
index f0e275d..be88402 100644
--- extract.py
+++ extract.py
@@ -3,2 +3,3 @@ import sys
     from spreadsheet import Spreadsheet
+from viewport import Viewport

@@ -8,3 +9,3 @@ if __name__ == '__main__':

-     sheet = Spreadsheet(filename)
+     sheet = Spreadsheet(filename, Viewport(2, 1, 12, 4))
```

```
diff --git spreadsheet.py spreadsheet.py
index 8a56f11..ff28dc3 100644
--- spreadsheet.py
+++ spreadsheet.py
@@ -7,3 +7,3 @@ from viewport import Viewport
     class Spreadsheet:
-         def __init__(self, filename):
+         def __init__(self, filename, viewport):
             book = xlrd.open_workbook(filename, formatting_info=True)
@@ -12,3 +12,3 @@ class Spreadsheet:
         self.sheet = sheet
-        self.vp = Viewport(2, 1, 12, 4)
+        self.vp = viewport

--
2.3.4
```

From 98606f7402ee7633dace6dfd5ac9d712ba23cb33 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 11:54:10 -0300
Subject: [PATCH 26/30] Add Spreadsheet.from_xls classmethod and refactor
Spreadsheet.__init__

diff --git extract.py extract.py
index be88402..f996a5b 100644

```
--- extract.py
+++ extract.py
@@ -9,3 +9,3 @@ if __name__ == '__main__':

- sheet = Spreadsheet(filename, Viewport(2, 1, 12, 4))
+ sheet = Spreadsheet.from_xls(filename, Viewport(2, 1, 12, 4))
```

diff --git spreadsheet.py spreadsheet.py
index ff28dc3..62f4a16 100644

```
--- spreadsheet.py
+++ spreadsheet.py
@@ -3,3 +3,2 @@ from itertools import chain
import xlrd
-from viewport import Viewport

@@ -7,3 +6,8 @@ from viewport import Viewport
class Spreadsheet:
- def __init__(self, filename, viewport):
+ def __init__(self, sheet, viewport):
+     self.sheet = sheet
+     self.vp = viewport
+
+ @classmethod
+ def from_xls(cls, filename, viewport):
+     book = xlrd.open_workbook(filename, formatting_info=True)
@@ -11,4 +15,3 @@ class Spreadsheet:

- self.sheet = sheet
- self.vp = viewport
+ return cls(sheet, viewport)
```

--
2.3.4

From 3a7961e537be4030257180cd1e0f0220e37cb0b2 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Tue, 2 Aug 2016 14:38:43 -0300
Subject: [PATCH 27/30] Make Viewport optional on instantiating Spreadsheet

```
diff --git spreadsheet.py spreadsheet.py
index 62f4a16..be8fe87 100644
--- spreadsheet.py
+++ spreadsheet.py
@@ -3,2 +3,3 @@ from itertools import chain
     import xlrd
+from viewport import Viewport

@@ -6,8 +7,12 @@ import xlrd
class Spreadsheet:
-    def __init__(self, sheet, viewport):
+    viewport = Viewport(0, 0, 1, 1)
+
+    def __init__(self, sheet, viewport=None):
        self.sheet = sheet
-        self.vp = viewport
+
+        if viewport:
+            self.viewport = viewport

    @classmethod
-    def from_xls(cls, filename, viewport):
+    def from_xls(cls, filename, viewport=None):
        book = xlrd.open_workbook(filename, formatting_info=True)
@@ -19,3 +24,3 @@ class Spreadsheet:
    def headers(self):
-        return [self.sheet.cell_value(i, j) for i, j in self.vp.rows[0]]
+        return [self.sheet.cell_value(i, j) for i, j in self.viewport.rows[0]]

@@ -24,3 +29,3 @@ class Spreadsheet:
    return [[self.sheet.cell_value(i, j) for i, j in indexes]
-        for indexes in self.vp.rows[1:]]
+        for indexes in self.viewport.rows[1:]]

--
2.3.4
```

From c8090732d9f31ee5314466deaa0fb87e9d888614 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Tue, 2 Aug 2016 14:44:03 -0300
Subject: [PATCH 28/30] Add CardStatement class to represent our input xls

```
diff --git extract.py extract.py
index f996a5b..9be9e38 100644
--- extract.py
+++ extract.py
@@ -2,4 +2,3 @@ import sys

-from spreadsheet import Spreadsheet
-from viewport import Viewport
+from statement import CardStatement

@@ -9,5 +8,5 @@ if __name__ == '__main__':
-    sheet = Spreadsheet.from_xls(filename, Viewport(2, 1, 12, 4))
+    stm = CardStatement.from_xls(filename)

-    for row in sheet:
+    for row in stm:
        print('{3: >6} {2}'.format(*row))
diff --git statement.py statement.py
new file mode 100644
index 0000000..607728e
--- /dev/null
+++ statement.py
@@ -0,0 +1,6 @@
+from spreadsheet import Spreadsheet
+from viewport import Viewport
+
+class CardStatement(Spreadsheet):
+    viewport = Viewport(2, 1, 12, 4)
+
+
+2.3.4
```

From 694c385d537c0e25c3482394149e7522e9d44574 Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sat, 30 Jul 2016 12:24:55 -0300
Subject: [PATCH 29/30] CardStatement.__str__

```
diff --git extract.py extract.py
index 9be9e38..3664cf1 100644
--- extract.py
+++ extract.py
@@ -10,3 +10,2 @@ if __name__ == '__main__':

-     for row in stm:
-         print('{3: >6} {2}'.format(*row))
+     print(str(stm))
diff --git statement.py statement.py
index 607728e..79e6531 100644
--- statement.py
+++ statement.py
@@ -6 +6,7 @@ class CardStatement(Spreadsheet):
     viewport = Viewport(2, 1, 12, 4)
+
+     def __str__(self):
+         out = []
+         for row in self:
+             out.append('{3: >6} {2}'.format(*row))
+         return '\n'.join(out)
--
2.3.4
```

From c866cc7e7252df6f6dc6b13368c4beaf8da7822e Mon Sep 17 00:00:00 2001
From: Henrique Bastos <henrique@bastos.net>
Date: Sun, 31 Jul 2016 13:43:23 -0300
Subject: [PATCH 30/30] Refactor CardStatement.__str__

```
diff --git statement.py statement.py
index 79e6531..268608c 100644
--- statement.py
+++ statement.py
@@ -8,5 +8,2 @@ class CardStatement(Spreadsheet):
     def __str__(self):
-        out = []
-        for row in self:
-            out.append('{3: >6} {2}'.format(*row))
-        return '\n'.join(out)
+        return '\n'.join('{3: >6} {2}'.format(*row) for row in self)
--
2.3.4
```