EventTools and CSS

EventTools generates plain HTML. Formatting is done via Cascading Style Sheets (CSS) only, not in the EventTools PHP code.

This note describes conventions in the EventTools HTML that can be useful when defining CSS to do desired formatting.

The best way to understand the HTML that EvenTools is generating is to look at the HTML using "Display Page Source" in your browser.

Design Goals

1. EventTools output should co-exist with other structures on HTML pages. Since EventTools doesn't generate pages, but rather pieces to put inside other pages, this allows you to format EventTools output separately from the rest of the page.

2. You might want to have multiple EventTools tables on the same page, formatted differently. This can be done via local CSS, but we want to enable a more global approach through a naming structure.

EventTools CSS naming structure

EventTools gives names to many elements in its output. The convention matches up with the design goal:

1. Event name starts with "et-" to distinguish EventTools-specific ones from everything else
2. The 2nd element is an abbreviation for the routine which wrote the table
3. The 3rd is the type
4. The 4th is an element identifier.

There are three types of classes: geographic, functional, and content:

- Geographic classes are used to represent where something is: The first row of a table entry. The left column of a table.
- Functional classes are used to represent the functional meaning of an element: The tour name. The layout owner's name.
- Content classes are used to identify something specific about the content of an element: Warning, required content not provided. Error, value not found.

A given element may be in several classes. For example, layout name element in a table will have both geographic class (so you can color e.g. the 2nd column of the table) and a functional class (so you can bold-face the layout name).

Example: A column in a table

et-faobd-col-owner
"et" starts all EventTools names
"faobd" is from the writing routine: format_all_ops_by_day
"col" means this is a column element
"owner" is the specific column. Some are named and some are numbered, depending on expected use.

Example: A particular cell class

et-faobd-cell-owner
"et" starts all EventTools names
"faobd" is from the writing routine: format_all_ops_by_day
"cell" means this is a column element
"owner" is the specific column

Element Conventions

Note that not all HTML output fully conforms to this yet; we're working it!

Table structure:

In addition to the <table> tag, we provide <thead>, <tbody> and <tfoot> tags in case you want to provide e.g. scrolling support with fixed header and footer.

Columns:

Neither "display" nor "visibility seem to make a column go away.

The first of these lines works; the next two don't:

```
.et-faobd-col-3 { background: #808080; }

.et-faobd-col-3 { display: none; }

.et-faobd-col-3 { visibility: collapse; }
```

To fix this, we provide spans on the elements: et-faobd-th-3 and et-faobd-td-3, so you can turn off a column with:

```
.et-faobd-th-4 { display: none; }
.et-faobd-td-4 { display: none; }
```

Remember the general CSS rule: any style on a row or cell overrules a column style.

Rows:

We provide CSS support for coloring rows alternately via:

```
<tr class=et-faobd-th"> (for the heading)
<tr>
<tr class="altrow">
<tr>
```

so you can e.g. apply colors to alternating rows.

Cells:

Note: Spans within a cell don't change the visibility of the cell; those outside do.

Sometimes you want to color by status, etc. We therefore provide (for cells where that's useful) spans like:

```
        <span class="et-faobd-session-status-60">
```

inside the cell itself.