# Rethinking Interaction with Literate Computing

Clemens Nylandsted Klokmose & Roman Rädle
Aarhus University

A *probability distribution* defines how likely different states of a random variable are. The probability distribution of a random variable $X$ is written:

$$P(X)$$

and is shorthand for the expression

$$Pr(X = x),$$

i.e. that the variable $X$ takes on the specific value $x$.

Random variables can be continuous (e.g. the height of a person) or discrete (the value showing on the face of a dice). The distribution of a discrete variable is described with a *probability mass function* (PMF) which gives each outcome a specific value. A continuous variable has a *probability density function* (PDF) which specifies the spread of the probability as a continuous function.

A probability distribution must assign probabilities in the range 0 (impossible) to 1 (definite) and the PMF or PDF **must** sum/integrate to exactly 1 as the random variable under consideration must take on *some* value.

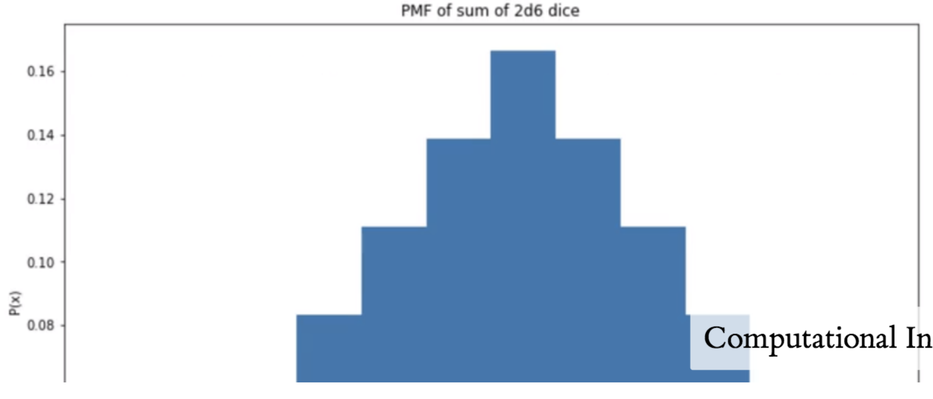
## PMF example: sum of dice rolls

A very simple discrete PMF is the expected value of the sum of two six-faced die. In this case, $P(X) = P(D_1 + D_2)$ for two uniform discrete variables $D_1, D_2 \in \{1, 2, 3, 4, 5, 6\}$

In [8]:
```
# the PMF of the sum of two dice rolls
def two_dice():
    # form the sum of the cross product of these possibilities
    roll_two = [i+j for i in range(1,7) for j in range(1,7)]
    # now plot the histogram
    pmf, edges, patches = plt.hist(roll_two, normed=True, bins=range(1,14))
    print("Sum of PMF %.2f" % np.sum(pmf)) # sum of probability should be *exactly* 1.0
    plt.title("PMF of sum of 2d6 dice")
    plt.xlabel("Sum of rolls x")
    plt.ylabel("P(x)")
```

In [9]:
```
two_dice()
```

```
Sum of PMF 1.00
```



# literate computing

is the act of weaving *"a narrative directly into a live computation, interleaving text with code and results to construct a complete piece that relies equally on the textual explanations and the computational components."*

Fernando Perez — co-creator of the popular Jupyter Notebook
http://blog.fperez.org/2013/04/literate-computing-and-computational.html

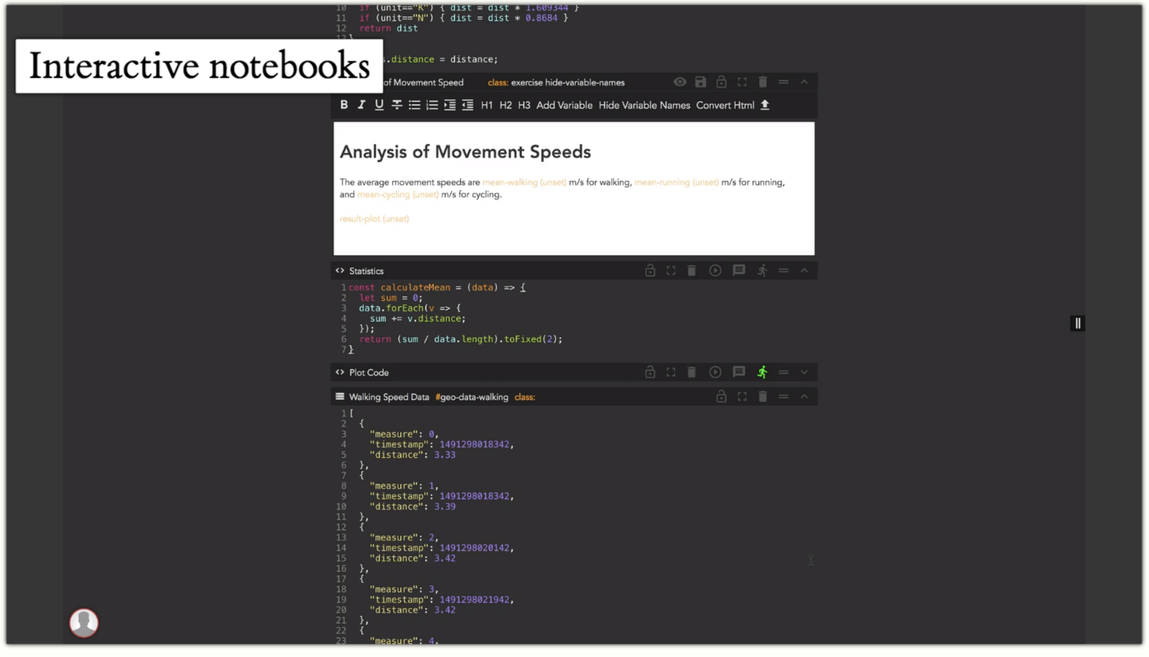
# Codestrates combines Literate Computing with Webstrates

Collaborative computation, extension, and development of applications

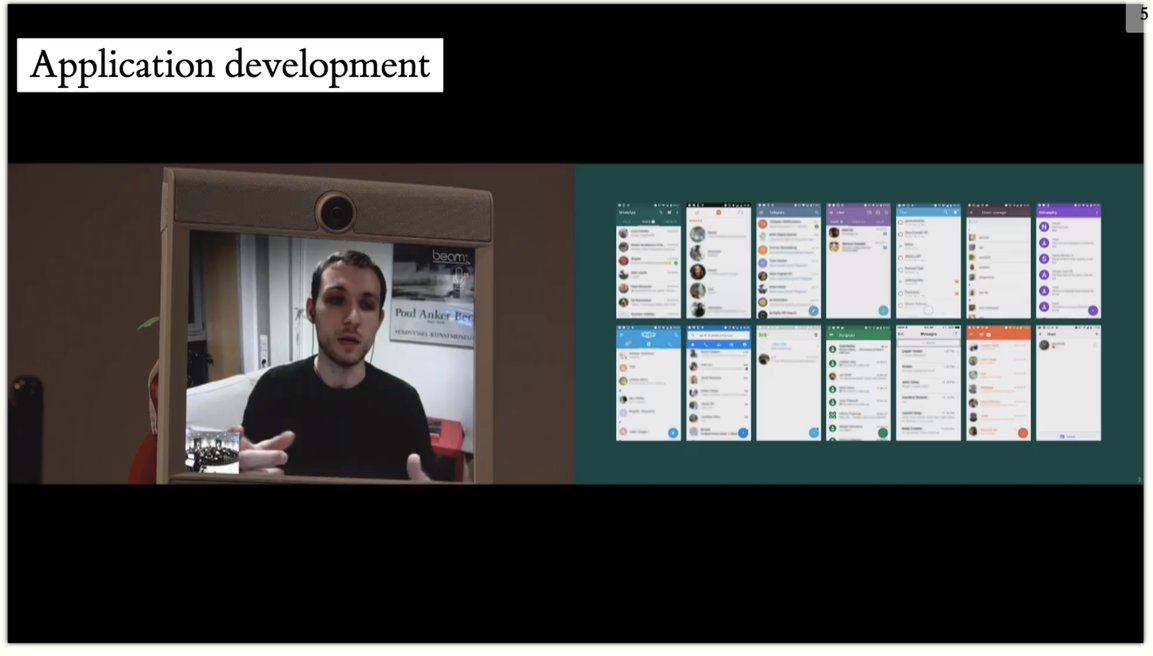
Prototyping similar to code playgrounds, but with potential to become usable applications with persistent states

Development environment for Webstrates that goes beyond paradigmatic application-document model

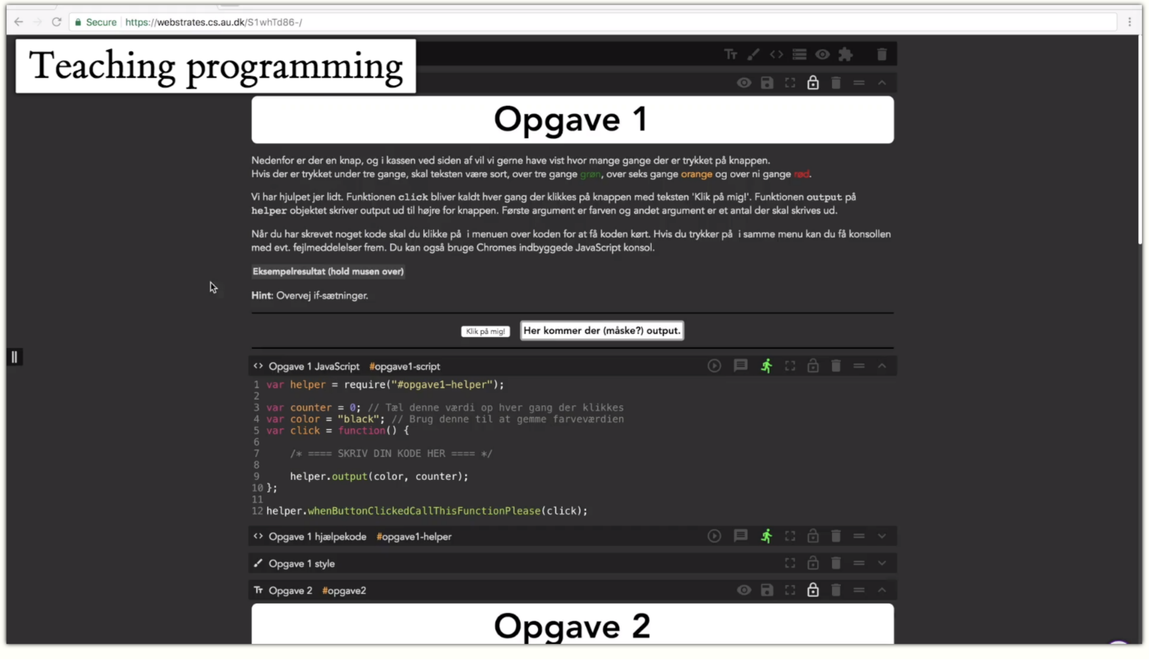
## Interactive notebooks




## Application development



## Teaching programming




## Physical computing