

Nov 07, 16 0:17	hw06-LET-output	Page 1/6
1	[dshin@acacia let]\$ Let /home/pl/hw06/tests/const-negative	
2	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
3	> % =====	
4	% negative constant. (value=-27)	
5		
6	-27	
7		
8	% =====	
9	-27	
10	>	
11	[dshin@acacia let]\$ Let /home/pl/hw06/tests/const-positive	
12	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
13	> % =====	
14	% positive constant. (value=12)	
15		
16	12	
17		
18	% =====	
19	12	
20	>	
21	[dshin@acacia let]\$ Let /home/pl/hw06/tests/diff-1	
22	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
23	> % =====	
24	% diff, where env=[i=1,v=5,x=10]. (value=9)	
25		
26	-(x,1)	
27		
28	% =====	
29	9	
30	>	
31	[dshin@acacia let]\$ Let /home/pl/hw06/tests/diff-2	
32	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
33	> % =====	
34	% diff, where env=[i=1,v=5,x=10]. (value=3)	
35		
36	-(1,-2)	
37		
38	% =====	
39	3	
40	>	
41	[dshin@acacia let]\$ Let /home/pl/hw06/tests/diff-error-1	
42	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
43	> % =====	
44	% diff, where env=[i=1,v=5,x=10]. (value=error)	
45	-(zero?(0),x)	
46		
47	% =====	
48	expval->num: arg=#(struct:bool-val #t)	
49	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/value.scm:13:17	
50	: (eopl:error (quote expval->num) "arg=~a" val)	
51	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:37:30:	
52	(- (expval->num (value-of expl env)) (expval->num (value-of exp2 env)))	
53	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:37:21:	
54	(num-val (- (expval->num (value-of expl env)) (expval->num (value-of exp2 env))))	
55	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:23:4: (	
56	let-values (((val) (value-of-1 exp env))) (trace-run-exit val) val)	
57	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:28: (	
58	expval->string (run string))	
59	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:15: (	
60	format "~a" (expval->string (run string)))	
61	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:6: (d	
62	isplay (format "~a" (expval->string (run string))))	
63	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:5:4: (l	
64	et-values (((string) (readfile file))) (display string) (display (format "~a" ((	
65	....) ....))) (newline))	

Nov 07, 16 0:17	hw06-LET-output	Page 2/6
57		
58		
59	=== context ===	
60	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:28:2: v	
61	alue-of-1	
62	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:21:2: v	
63	alue-of	
64	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:4:2: ru	
65	nfile	
66		
67	>	
68	[dshin@acacia let]\$ Let /home/pl/hw06/tests/diff-error-2	
69	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
70	> % =====	
71	% diff, where env=[i=1,v=5,x=10]. (value=error)	
72		
73	-(x,zero?(0))	
74		
75	% =====	
76	expval->num: arg=#(struct:bool-val #t)	
77	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/value.scm:13:17	
78	: (eopl:error (quote expval->num) "arg=~a" val)	
79	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:37:30:	
80	(- (expval->num (value-of expl env)) (expval->num (value-of exp2 env)))	
81	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:37:21:	
82	(num-val (- (expval->num (value-of expl env)) (expval->num (value-of exp2 env))))	
83	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:23:4: (	
84	let-values (((val) (value-of-1 exp env))) (trace-run-exit val) val)	
85	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:28: (	
86	expval->string (run string))	
87	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:15: (	
88	format "~a" (expval->string (run string)))	
89	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:6: (d	
90	isplay (format "~a" (expval->string (run string))))	
91	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:5:4: (l	
92	et-values (((string) (readfile file))) (display string) (display (format "~a" ((	
93	....) ....))) (newline))	
94		
95	=== context ===	
96	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:28:2: v	
97	alue-of-1	
98	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:21:2: v	
99	alue-of	
100	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:4:2: ru	
101	nfile	
102		
103	>	
104	[dshin@acacia let]\$ Let /home/pl/hw06/tests/diff-nested-left	
105	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
106	> % =====	
107	% nested diff. (-11)	
108		
109	-(-(44,33),22)	
110		
111	% =====	
112	-11	
113	>	
114	[dshin@acacia let]\$ Let /home/pl/hw06/tests/diff-nested-right	
115	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
116	> % =====	
117	% nested diff. (44)	
118		
119	-(55,-(22,11))	
120		

Nov 07, 16 0:17	hw06-LET-output	Page 3/6
107	% =====	
108	44	
109	>	
110	[dshin@acacia let]\$ Let /home/pl/hw06/tests/if-error	
111	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
112	> % =====	
113	% if with non bool checking. (error)	
114		
115	if 1 then 2 else 3	
116		
117	% =====	
118	expval->bool: arg=(struct:num-val 1)	
119	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/value.scm:19:17	
120	: (eopl:error (quote expval->bool) "arg=~a" val)	
121	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:40:19:	
122	(if (expval->bool (value-of expl env)) (value-of exp2 env) (value-of exp3 env))	
123	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:23:4: (	
124	let-values (((val) (value-of-1 exp env))) (trace-run-exit val) val)	
125	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:28: (	
126	expval->string (run string))	
127	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:15: (	
128	format "~a" (expval->string (run string)))	
129	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:6: (d	
130	isplay (format "~a" (expval->string (run string))))	
131	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:5:4: (l	
132	et-values (((string) (readfile file))) (display string) (display (format "~a" ((	
133	....) ....))) (newline))	
134		
135	=== context ===	
136	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:28:2: v	
137	alue-of-1	
138	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:21:2: v	
139	alue-of	
140	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:4:2: ru	
141	nfile	
142		
143	>	
144	[dshin@acacia let]\$ Let /home/pl/hw06/tests/if-false-1	
145	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
146	> % =====	
147	% if conditional. (value=4)	
148		
149	if zero?(1) then 3 else 4	
150		
151	% =====	
152	3	
153	>	
154	[dshin@acacia let]\$ Let /home/pl/hw06/tests/if-true-1	
155	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
156	> % =====	
157	% if conditional. (value=3)	
158		
159	if zero?(0) then 3 else 4	
160		
161	% =====	

Nov 07, 16 0:17	hw06-LET-output	Page 4/6
162	3	
163	>	
164	[dshin@acacia let]\$ Let /home/pl/hw06/tests/if-true-2	
165	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
166	> % =====	
167	% if conditional where true exp is not evaluated. (value=3)	
168		
169	if zero?(1) then foo else 3	
170		
171	% =====	
172	3	
173	>	
174	[dshin@acacia let]\$ Let /home/pl/hw06/tests/let	
175	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
176	> % =====	
177	% let. (value=3)	
178		
179	let x=3	
180	in x	
181		
182	% =====	
183	3	
184	>	
185	[dshin@acacia let]\$ Let /home/pl/hw06/tests/let-body	
186	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
187	> % =====	
188	% make sure the let body is evaluated. (value=7)	
189		
190	let x=8 in -(x,1)	
191		
192	% =====	
193	7	
194	>	
195	[dshin@acacia let]\$ Let /home/pl/hw06/tests/let-nested	
196	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
197	> % =====	
198	% nested let make lexical scoping. p66. (value=3)	
199		
200	let z=5	
201	in let x=3	
202	in let y=-(x,1)	
203	in let x=4	
204	in -(z,-(x,y))	
205		
206	% =====	
207	3	
208	>	
209	[dshin@acacia let]\$ Let /home/pl/hw06/tests/let-shadow-body	
210	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
211	> % =====	
212	% shadowing in let. (value=4)	
213		
214	let x=3	
215	in let x=4	
216	in x	
217		
218	% =====	
219	4	
220	>	
221	[dshin@acacia let]\$ Let /home/pl/hw06/tests/let-shadow-rhs	
222	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
223	> % =====	
224	% shadowing in let. (value=2)	
225		
226	let x=3	
227	in let x=-(x,1)	

Nov 07, 16 0:17	hw06-LET-output	Page 5/6
228	in x	
229		
230	% =====	
231	2	
232	>	
233	[dshin@acacia let]\$ Let /home/pl/hw06/tests/let-var	
234	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
235	> % =====	
236	% make sure the let var is evaluated. (value=7)	
237		
238	let x=-(8,1) in x	
239		
240	% =====	
241	7	
242	>	
243	[dshin@acacia let]\$ Let /home/pl/hw06/tests/variable	
244	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
245	> % =====	
246	% variable, where env=[i=1,v=5,x=10]. (value=10)	
247		
248	x	
249		
250	% =====	
251	10	
252	>	
253	[dshin@acacia let]\$ Let /home/pl/hw06/tests/variable-error-1	
254	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
255	> % =====	
256	% unbound variable, where env=[i=1,v=5,x=10]. (value=error)	
257		
258	foo	
259		
260	% =====	
261	apply-env: undefined foo	
262	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/env.scm:13:22:	
263	(eopl:error (quote apply-env) "undefined ~a" search-sym)	
264	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:23:4: (	
265	let-values (((val) (value-of-1 exp env))) (trace-run-exit val) val)	
266	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:28: (	
267	expval->string (run string))	
268	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:15: (	
269	format "~a" (expval->string (run string)))	
270	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:6: (d	
271	isplay (format "~a" (expval->string (run string))))	
272	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:5:4: (l	
273	et-values (((string) (readfile file))) (display string) (display (format "~a" ((	
274	....) ....))) (newline))	
275		
276	=== context ===	
277	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:21:2: v	
278	alue-of	
279	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:4:2: ru	
280	nfile	
281		
282	>	
283	[dshin@acacia let]\$ Let /home/pl/hw06/tests/variable-error-2	
284	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
285	> % =====	
286	% unbound variable, where env=[i=1,v=5,x=10]. (value=error)	
287		
288	-(x,foo)	
289		
290	% =====	
291	apply-env: undefined foo	
292	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/env.scm:13:22:	

Nov 07, 16 0:17	hw06-LET-output	Page 6/6
285	(eopl:error (quote apply-env) "undefined ~a" search-sym)	
286	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:23:4: (	
287	let-values (((val) (value-of-1 exp env))) (trace-run-exit val) val)	
288	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:38:33:	
289	(expval->num (value-of exp2 env))	
290	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:37:30:	
291	(- (expval->num (value-of exp1 env)) (expval->num (value-of exp2 env)))	
292	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:37:21:	
293	(num-val (- (expval->num (value-of exp1 env)) (expval->num (value-of exp2 env))))	
294		
295	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:23:4: (	
296	let-values (((val) (value-of-1 exp env))) (trace-run-exit val) val)	
297	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:28: (	
298	expval->string (run string))	
299	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:15: (	
300	format "~a" (expval->string (run string)))	
301	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:7:6: (d	
302	isplay (format "~a" (expval->string (run string))))	
303	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:5:4: (l	
304	et-values (((string) (readfile file))) (display string) (display (format "~a" ((	
305	....) ....))) (newline))	
306		
307	=== context ===	
308	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:21:2: v	
309	alue-of	
310	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:28:2: v	
311	alue-of-1	
312	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:21:2: v	
313	alue-of	
314	/home/dshin/2012-ea0011-pl/eopl3-programs/eopl3-interpreters/let/run.scm:4:2: ru	
315	nfile	
316		
317	>	
318	[dshin@acacia let]\$ Let /home/pl/hw06/tests/zero-false	
319	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
320	> % =====	
321	% zero? (value=#f)	
322		
323	zero?(37)	
324		
325	% =====	
326	#f	
327	>	
328	[dshin@acacia let]\$ Let /home/pl/hw06/tests/zero-true	
329	Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc.	
330	> % =====	
331	% zero? (value=#t)	
332		
333	zero?(0)	
334		
335	% =====	
336	#t	
337	>	
338	[dshin@acacia let]\$	